

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**ESCUELA DE POSGRADO**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**UNIDAD DE POSGRADO**

**“MARCO METODOLÓGICO DEL PROCESO DE  
VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE PARA  
PEQUEÑAS Y MEDIANAS EMPRESAS”**

**PARA OBTENER EL GRADO ACADÉMICO DE MAGISTER EN INGENIERÍA DE  
SISTEMAS E INFORMÁTICA**

**CON MENCIÓN EN DIRECCIÓN Y GESTIÓN DE TECNOLOGÍA DE  
INFORMACIÓN**

**AUTOR**

Lizette Grados Aguirre

Lima – Perú

2015

Tesis presentada a la Unidad de Post Grado  
de la Facultad Ingeniería de Sistemas e  
Informática de la UNMSM para obtener el  
grado de Magister en Ingeniería de  
Sistemas e Informática

**Orientador:** Dra. Nora La Serna Palomino



© Lizette Grados Aguirre 2015.  
Todos los derechos reservados.

Este trabajo está dedicado a Dios y a toda mi familia, en especial a mis padres, quienes me han enseñado con el ejemplo que no hay nada imposible si uno se esfuerza por lograrlo.

## **AGRADECIMIENTOS**

A Dios por guiarme y regalarme oportunidades en toda mi vida.

A mis Padres por el apoyo incondicional y el amor que me regalan todos los días.

A la Dra. Nora la Serna Palomino, por su orientación, dedicación y apoyo para que este trabajo de investigación cumpla con los objetivos trazados.

A los profesores de la UNMSM, principalmente al Dr. David Mauricio Sánchez, por sus enseñanzas que me sirvieron de mucho.

A Kurt Alfaro Hoffman y Alex Vidaurre More que me ayudaron con su experiencia y aportes a culminar este trabajo.

# MARCO METODOLÓGICO DEL PROCESO DE VERIFICACIÓN Y VALIDACIÓN PARA PEQUEÑAS Y MEDIANAS EMPRESAS

## RESUMEN

Hoy en día es complicado para una Pyme (pequeña casa de desarrollo de Software) aplicar los Modelos de Calidad y Pruebas propuestos tales como: CMMI, CMM, MOPROSOFT, COMPETISOFT, TMMi, TMM, TPI, TMap que hacen referencia al proceso de V&V, esto principalmente por: 1) el grado de complejidad, la mayoría de estos modelos han sido estructurados para ser aplicados en empresas grandes; 2) las restricciones de costo porque las Pymes no tienen el suficiente dinero para contratar a los especialistas que puedan brindarles las directivas que necesitan o 3) simplemente por el desconocimiento que se tiene sobre los beneficios que obtendrían.

El presente trabajo de investigación tiene como principal objetivo elaborar un **Marco Metodológico del Proceso de Verificación y Validación** ajustado a la realidad socio económico de las Pymes y enfocado en los procesos claves de V&V, por ello, se tomaron como referencia los modelos y estándares de calidad: IEEE Std. 1012, IEEE Std. 829, IEEE Std. 1008, en el trabajo se muestra una guía metodológica, los roles y competencias del equipo de trabajo, plantillas/listas de verificación base y las principales métricas.

**Palabras clave:** Calidad, V&V, Pruebas, CMMI, CMM, MOPROSOFT, COMPETISOFT, TMMi, TMM, TPI, TMap, IEEE Std. 1012, IEEE Std. 829, IEEE Std. 1008.

## **METHODOLOGICAL FRAMEWORK OF VERIFICATION AND VALIDATION PROCESS FOR SMALL AND MEDIUM ENTERPRISES**

### **ABSTRACT**

Today it is difficult for an SME (small house software development) apply Quality Models and proposed tests such as CMMI, CMM, MOPROSOFT, COMPETISOFT, TMMi, TMM, TPI, TMap referring to the process of V & V, this mainly by: 1) the degree of complexity, most of these models have been structured to be applied in large companies; 2) cost constraints because SMEs do not have enough money to hire experts who can provide the policies that need or 3) simply by ignorance we have about the benefits they would get.

The present research has as main objective to develop a methodological framework Process Verification and Validation adjusted for socio-economic reality of SMEs and focused on the key processes of V & V, therefore, were taken as reference models and quality standards: IEEE Std 1012, IEEE Std 829, IEEE Std 1008, work a methodological guide, the roles and responsibilities of the team, templates / lists based verification and key metrics shown.

**Keywords:** Quality, Verification and Validation, Testing, CMMI, CMM, MOPROSOFT, COMPETISOFT, TMMi, TMM, TPI, TMap, IEEE Std 1012, IEEE Std 829, IEEE Std 1008

# ÍNDICE

<b>CAPÍTULO 1: INTRODUCCIÓN.....</b>	<b>14</b>
1.1 Antecedentes del Problema.....	14
1.2 Definición del Problema.....	16
1.3 Objetivos de Investigación.....	17
1.3.1 Objetivo General .....	17
1.3.2 Objetivos Específicos .....	17
1.4 Motivación del Estudio.....	17
1.5 Alcance de la Investigación .....	19
1.6 Organización de la Tesis .....	19
<b>CAPÍTULO 2: MARCO TEÓRICO.....</b>	<b>21</b>
2.1 Evolución del proceso de Verificación y Validación.....	21
2.2 Aseguramiento de Calidad de Software Vs Control de Calidad de Software .....	29
2.2.1 Surgimiento de SQA (Software Quality Assurance) .....	29
2.2.2 Definición de SQA (Software Quality Assurance) .....	31
2.2.3 SQA no es lo mismo que SQC (Software Quality Control).....	32
2.2.4 Esquema Conceptual del proceso de Verificación y Validación.....	33
2.3 Modelo V de Verificación y Validación.....	35
2.4 Estándares de Verificación y Validación.....	39
2.4.1 ISO / IEC 29110 .....	41
<b>CAPÍTULO 3: ESTADO DEL ARTE DEL PROCESO VERIFICACIÓN Y VALIDACIÓN PARA PYMES .....</b>	<b>47</b>
3.1 Modelos para el proceso de Verificación y Validación.....	47
3.1.1 COMPETISOFT .....	47
3.1.2 CMMI for Development .....	56
3.1.3 TMM - Test Maturity Model .....	60
3.1.4 TMMi - Test Maturity Model Integration.....	64

3.1.5	TMap - Test Management Approach .....	68
3.1.6	MoProSoft .....	79
<b>3.2</b>	<b>Comparación de Modelos de Calidad .....</b>	<b>82</b>
<b>3.3</b>	<b>Artículos relacionados .....</b>	<b>86</b>
3.3.1	“Integration, Verification, Validation, Test And Evaluation (IVVT&E) Framework for System of Systems (SoS)” .....	86
3.3.2	Metodología para implantar el Modelo Integrado de Capacidad de Madurez en grupos pequeños y emergentes. ....	87
3.3.3	Developing an analytical model for planning systems verification, validation and testing processes .....	89
3.3.4	25 años de Investigacion Testing en Brazil. ....	90
3.3.5	A systematic mapping study on the combination of static and dynamic quality assurance techniques .....	93
3.3.6	Assessment of software quality: Choquet Integral Approach.....	96
3.3.7	QRP: a CMMI Appraisal Tool for Project Quality Management.....	97
 <b>CAPÍTULO 4: MARCO METODOLÓGICO PROPUESTO DEL PROCESO DE V&amp;V DE SOFTWARE PARA LAS PYMES EN EL PERÚ ..... 100</b>		
<b>4.1</b>	<b>Roles del proceso de Verificación y Validación de Software .....</b>	<b>100</b>
<b>4.2</b>	<b>Responsabilidades de las áreas que participan en el proceso de Verificación y Validación de Software .....</b>	<b>102</b>
<b>4.3</b>	<b>Descripción del Marco Metodológico Propuesto para el proceso de Verificación y Validación de Software .....</b>	<b>104</b>
4.3.1	Fase de Planeación del Proceso de Verificación y Validación de Software.....	108
4.3.2	Fase de Análisis del Proceso de Verificación y Validación de Software .....	112
4.3.3	Fase de Diseño del Proceso de Verificación y Validación de Software .....	115
4.3.4	Fase de Pruebas y Ratificación del Proceso de Verificación y Validación de Software .....	117
<b>4.4</b>	<b>Gestión de Defectos para el Proceso de Verificación y Validación de Software.....</b>	<b>119</b>
<b>4.5</b>	<b>Indicadores del Proceso de Verificación y Validación de Software.....</b>	<b>120</b>
<b>4.6</b>	<b>Plantillas del Proceso de Verificación y Validación de Software.....</b>	<b>122</b>
4.6.1	Planeación .....	122
4.6.2	Análisis .....	137
4.6.3	Diseño.....	148

4.6.4	Pruebas y Ratificación .....	150
<b>CAPÍTULO 5: CASO DE ESTUDIO .....</b>		<b>153</b>
5.1	Características de la empresa .....	153
5.2	Instalación y esfuerzo del Marco Metodológico propuesto .....	153
5.3	Análisis del estado inicial de la empresa .....	154
5.4	Aplicación del Marco Metodológico del Proceso de Verificación y Validación.....	156
5.4.1	Planeación .....	156
5.4.2	Análisis .....	178
5.4.3	Diseño.....	187
5.4.4	Pruebas y Ratificación .....	196
5.5	Estadísticas y Métricas .....	199
5.5.1	Defectos por Módulo. ....	199
5.5.2	Numero de caminos probados.....	199
5.5.3	Efectividad de la detección de defectos.....	200
5.5.4	Presupuesto utilizado en revisiones. ....	200
5.5.5	Incidencias Por Tipo de Revisiones y Pruebas.....	201
5.6	Costos y tiempo.....	201
5.7	Resultados obtenidos. ....	202
<b>CAPÍTULO 6: CONCLUSIONES Y TRABAJOS FUTUROS .....</b>		<b>203</b>
6.1	Conclusiones .....	203
6.2	Trabajos Futuros .....	204
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>206</b>

# INDICE DE FIGURAS

Figura 2-1. Aplicación del Proceso de V&V en el ciclo de vida del SW (ISO/IEC 12207:1995).....	35
Figura 2-2 Modelo V del Proceso V&V (Pfleeger, 2001; Pressman, 2000). .....	36
Figura 2-3 Estructura ISO/IEC 29110 (ISO/IEC 29110, 2013) .....	44
Figura 3-1 Actividades de Pruebas en el tiempo .....	47
Figura 3-2 Estructura CMMI .....	57
Figura 3-3 Estructura de Área de Proceso.....	58
Figura 3-4 Representaciones Continua y Escalonada CMMI .....	59
Figura 3-5 Estructura de Nivel TMM .....	61
Figura 3-6 Niveles CMM.....	62
Figura 3-7 Niveles TMMI .....	65
Figura 3-8 Estructura Área de Proceso TMMI .....	67
Figura 3-9 Modelo TMAP .....	69
Figura 3-10 Diagrama de categorías de procesos Moprosoft.....	80
Figura 3-11 Diagrama de relación entre procesos Moprosoft.....	82
Figura 4-1 Esquema General: Fases del ciclo de vida del SW Vs. Proceso V&V (Elaboración Propia) .....	105
Figura 4-2 Fases del Marco Propuesto para el Proceso de V&V de Software (Elaboración Propia) .....	108
Figura 4-3 Fase de Planeación del Proceso de V&V de SW: Solicitud Inicial (Elaboración Propia) .....	109
Figura 4-4 Fase de Planeación del Proceso de V&V de SW: Estimación Inicial (Elaboración Propia) .....	111
Figura 4-5 Fase de Análisis del Proceso de V&V de SW: EEFF/EETT (Elaboración Propia).....	113
Figura 4-6 Fase de Diseño del Proceso de V&V de SW: Casos de Prueba (Elaboración Propia)	116
Figura 4-7 Fase de Pruebas/Ratificación del Proceso de V&V de SW (Elaboración Propia) .....	118
Figura 4-8 Indicadores utilizados en el Marco del proceso de V&V propuesto (Elaboración Propia) .....	120
Figura 5-1 Niveles de Escalamiento (Elaboración Propia) .....	175
Figura 5-2 Densidad de Incidencias por Aplicativo (Elaboración Propia).....	199



Figura 5-3 Incidencias Por Tipo de Revisiones y Pruebas (Elaboración Propia) .....	201
--	-----

# INDICE DE TABLAS

Tabla 2-1 Control de calidad vs. Aseguramiento de la calidad (Elaboración Propia) .....	33
Tabla 2-2 Estrategias de Análisis Estático (IEEE 610.12-1990; Pressman, 2000) .....	37
Tabla 2-3 Estrategias de Análisis Dinámico (Beizer, 1990; Kaner, 1999; Myer, 2004; Bach, 2001,2003; Pérez, 2007; Black, 2002) .....	39
Tabla 2-4 Normas/Estándares Calidad (ISO 9001,2000; ISO 9000,2006; IEEE Std 730-2002; IEEE Std 1028-1997).....	40
Tabla 2-5 IEEE 1012, 2014.....	41
Tabla 3-1 Cuadro comparativo de Procesos de Calidad enfocados al proceso de V&V (Elaboración Propia) .....	85
Tabla 4-2 Roles de Verificación y Validación (Elaboración Propia).....	102
Tabla 4-3 Fase de Planeación del Proceso de V&V de SW: Solicitud Inicial (Elaboración Propia)	110
Tabla 4-4 Fase de Planeación del Proceso de V&V de SW: Estimación Inicial (Elaboración Propia) .....	112
Tabla 4-5 Fase de Análisis del Proceso de V&V de SW: EEEF (Elaboración Propia) .....	114
Tabla 4-6 Fase de Análisis del Proceso de V&V de SW: Plan de V&V (Elaboración Propia) .....	115
Tabla 4-7 Fase de Diseño del Proceso de V&V de SW: Diseño de Pruebas (Elaboración Propia)	117
Tabla 4-8 Fase de Pruebas del Proceso de V&V de SW: Ejecución de Pruebas (Elaboración Propia) .....	119
Tabla 4-9 Plantilla de Solicitud Inicial para la Estimación (Elaboración Propia) .....	124
Tabla 4-10 Plantilla de Lista de Verificación para la Solicitud Inicial (Elaboración Propia) .....	128
Tabla 4-11 Plantilla para el Registro de Hallazgos de Solicitud Inicial (Elaboración Propia) .....	130
Tabla 4-12 Estimación Inicial: Factores Técnicos (Elaboración Propia).....	132
Tabla 4-13 Estimación Inicial: Factores de Ambiente (Elaboración Propia) .....	132
Tabla 4-14 <i>Calificación de Registro de Requerimientos</i> (Elaboración Propia) .....	133
Tabla 4-15 Modelo de Estimación Inicial (Elaboración Propia) .....	135
Tabla 4-16 puntos a considerar en Plan V&V (Elaboración Propia) .....	136
Tabla 4-17 Listas de verificación de Requisitos (Elaboración Propia).....	140
Tabla 4-18 Listas de verificación de Especificación Funcional (Elaboración Propia).....	144
Tabla 4-19 Lista de Verificación de Especificación Técnica (Elaboración Propia) .....	147

Tabla 4-20 Plantilla de Casos de Prueba (Elaboración Propia) .....	149
Tabla 4-21 Plantilla de Reporte de Defectos (Elaboración Propia) .....	151
Tabla 5-1 Esfuerzo realizado para la actividad de instalación (Elaboración Propia).....	154
Tabla 5-2 Esfuerzo realizado para la actividad de análisis (Elaboración Propia).....	155
Tabla 5-3 Proceso Actual de Desarrollo – Pruebas de QBOSS (Elaboración Propia) .....	155
Tabla 5-4 Esfuerzo realizado para la actividad de aplicación del Marco Propuesto (Elaboración Propia) .....	156
Tabla 5-5 Formato de Solicitud Inicial (Elaboración Propia) .....	158
Tabla 5-6. Listas de verificación de Solicitud Inicial (Elaboración Propia).....	161
Tabla 5-7 Registro de Hallazgos (Elaboración Propia).....	163
Tabla 5-8 Factores Técnicos (Elaboración Propia).....	164
Tabla 5-9 Factores de Ambiente (Elaboración Propia) .....	164
Tabla 5-10 Registro de Requerimientos (Elaboración Propia) .....	166
Tabla 5-11 Modelo de Estimación (Elaboración Propia).....	168
Tabla 5-12 Involucrados del Proyecto (Elaboración Propia).....	169
Tabla 5-13 Severidad de los Hallazgos (Elaboración Propia) .....	172
Tabla 5-14 Severidad de los defectos (Elaboración Propia).....	173
Tabla 5-15 Definición de la Estrategia (Elaboración Propia) .....	174
Tabla 5-16 Entregables (Elaboración Propia) .....	176
Tabla 5-17 Riesgos del Proceso de Certificación (Elaboración Propia) .....	177
Tabla 5-18 Revisión de Requisitos (Elaboración Propia) .....	182
Tabla 5-19 Revisión de Requerimientos Funcionales (Elaboración Propia) .....	185
Tabla 5-20 Revisión Técnica (Elaboración Propia).....	187
Tabla 5-21 Registro de Casos de Prueba (Elaboración Propia).....	195
Tabla 5-22 Registro de Defectos (Elaboración Propia) .....	198
Tabla 5-23 Propuesta de Costos y Duración de Capacitación y Asesorías (Elaboración Propia) .	201

# Capítulo 1: Introducción

## 1.1 Antecedentes del Problema

Actualmente la Calidad del Software constituye uno de los procesos más importantes en cualquier industria. Las empresas dedicadas a la implementación de Software en un afán por formalizar y optimizar sus procesos optan por incluir en su organización Modelos como: CMMI (Capability Maturity Model Integration), TMMi (Test Maturity Model Integration), TMM (Test Maturity Model), TPI (Test Process Improvement), TMap (Test Management Approach), COMPETISOFT o Moprosoft que centran sus esfuerzos en conducir a las organizaciones en producir servicios/productos de calidad. También se incluyen diferentes estándares relacionados a la Calidad de Software como: IEEE Std. 1012, IEEE Std. 829, IEEE Std. 1008, entre otros.

Al revisarlos vemos que están estructurados para ser aplicados en organizaciones grandes y medianas, lo que supone una gran inversión en dinero, tiempo y recursos. Los modelos son muy genéricos e indican qué se debe hacer, pero no indican cómo implementarlo, para los modelos relacionados a las **Pymes** (pequeña casa que fabrica Software) no están focalizados en el proceso de Verificación y Validación sino en todo el proceso de Desarrollo de Software, las Pymes tendrían que contratar expertos para entender qué procesos, métricas, roles, plantillas y listas de verificación son los necesarios para llevar a cabo un adecuado proceso de Verificación y Validación de Software, estas restricciones y limitaciones no permiten implementar un software de calidad en las pequeñas empresas.

Por otro lado, durante la implementación del software en una Pyme, el Control de Calidad enfocado en los procesos de Verificación y Validación de Software, muchas veces no es considerado o se incluye demasiado tarde, principalmente por las siguientes razones:

**1. Debido a las restricciones de tiempo y costo el esfuerzo dedicado al proceso de Verificación y Validación de Software es nulo, escaso o tardío.**

Un estudio realizado por Tokar y Mankeforts destaca que el 60% de los desarrolladores encuestados afirma que las actividades de Verificación y Validación son las primeras que abandonan cuando se encuentran bajo presión (TOKAR R., MANKEFORTS S., 2003). En los últimos años, han surgido diferentes modelos de referencia de procesos cuyo propósito es proporcionar las guías adecuadas para que las organizaciones pongan en marcha iniciativas de mejora de procesos que permitan incrementar la calidad tanto de sus procesos como de sus productos. Sin embargo, son todavía numerosas las empresas que continúan sin invertir los recursos necesarios (tiempo, personas, infraestructura) para llevar a cabo satisfactoriamente mejoras dirigidas al proceso de Verificación y Validación (TASSEY, G. 2002). Generalmente, la Verificación no comienza hasta que el producto no se encuentra casi finalizado.

**2. No se cuenta con un modelo de competencias relacionado con cada uno de los perfiles profesionales.** Esta situación se agrava en pequeñas y medianas empresas (PYMES) donde, habitualmente, no se dispone de recursos debidamente cualificados y específicamente dedicados a la realización de las actividades relacionadas con el proceso de Verificación y Validación de Software (MISHRA, D., MISHRA, A. 2007), a veces las actividades de Verificación y Validación, se llevan a cabo por el propio desarrollador, al encontrarse en la fase final del proyecto la presión del tiempo es muy fuerte, por lo que el tiempo que se invierte en la ejecución de la Verificación es mínimo. Este entorno caótico y un proceso de Verificación y Validación pobre e incontrolado conllevan obtener un sistema con muchos defectos y un producto carente de calidad; lo cual originará una gran carga de retrabajo, con su consecuente incremento de los costos y la insatisfacción del cliente.

**3. Dificultad en la implementación de los procesos de mejora.** Es importante resaltar que la preparación previa a la acreditación de estos estándares es larga y costosa, porque los modelos de mejora, proceso y

evaluación, propuestos por el SEI e ISO, están estructurados para ser aplicables a grandes empresas (SAEDIAN, 1979). Diversos estudios presentan que la aplicación de éstos es difícil para las PYMES debido a que un proyecto de mejora siguiendo un modelo a grandes organizaciones supone una gran inversión en dinero, tiempo y recursos, además las recomendaciones son complejas de aplicar y el retorno de la inversión se produce a largo plazo (BATISTA, J Y A. FIGUEREIDO, 2000). Además, en las PYMES la aplicación de estos modelos se agrava aún más ya que existe un problema “cultural” cuando se quiere importar y adoptar, sin más, modelos creados para otro tipo de organizaciones, ya que, como señala (ZAHARAN, 1998) si el proceso no es “compatible” con la cultura de la organización será rechazado por el “cuerpo” organizacional como sucede con los trasplantes de órganos.

Por ello, en el presente trabajo de investigación se propone un Marco Metodológico que pueda ser utilizado por las Pymes y les permita emprender un proceso de Verificación y Validación de software adecuado desde la fase inicial hasta su post implantación. Asimismo se incluirá dentro del Marco Metodológico aquellas competencias relacionadas con el desempeño de las tareas de verificación y validación que son imprescindibles para la gestión del equipo.

Para medir los resultados del Marco Metodológico, se seleccionó el proyecto de **Sistema de Control de Inventarios** de la empresa **QBOSS SAC**, se analizó la situación de la Pyme antes de la implementación, luego se aplicó el Marco Metodológico propuesto y finalmente se muestran los resultados positivos que obtuvo la organización.

## **1.2 Definición del Problema**

Por lo descrito anteriormente, no existe un Marco Metodológico del proceso de Verificación y Validación para una Pyme que se dedica al desarrollo de Software que brinde los lineamientos detallados e indique cómo llevar a cabo un proceso de control de calidad eficiente.

### 1.3 Objetivos de Investigación

#### 1.3.1 Objetivo General

- Elaborar un **Marco Metodológico del Proceso de Verificación y Validación de software** para una Pyme que se dedica al desarrollo de Software basado en los modelos y estándares de calidad más representativos del mercado. Asimismo, se incluirá una guía metodológica detallada, los roles y competencias del equipo de trabajo, las principales métricas y las plantillas/listas de verificación base que deberían ser considerados durante la implementación de un Software de Calidad.

#### 1.3.2 Objetivos Específicos

- Realizar un estudio de los principales modelos, estándares, normas y estudios relacionados con el Proceso de Verificación y Validación de Software.
- Realizar comparaciones entre de los Modelos de Calidad y determinar qué ventajas y deficiencias presentan para ser aplicados en una Pyme que se dedica al desarrollo de Software.
- Identificar y definir un conjunto de roles y competencias que debería tener el equipo de trabajo para asegurar la correcta ejecución de los procesos de Verificación y Validación de Software.
- Definir las principales métricas, plantillas y listas de verificación que servirán como base para la ejecución de los procesos de Verificación y Validación de Software.
- Realizar una aplicación del Marco metodológico propuesto a una Pyme que se dedica al desarrollo de Software.

### 1.4 Motivación del Estudio

- Actualmente existe una amplia literatura sobre Calidad de Software, pero los modelos, estándares y normas existentes enfocadas en el proceso de Verificación y Validación de Software presentan las siguientes restricciones:
  - Son complejos de implementar, te dicen qué se debe hacer, pero no te indican cómo puedes implementarlo.

- Requieren de una fuerte inversión de dinero y tiempo para implementar todo el proceso descrito en los modelos.
  - No se identifica qué Roles/Perfiles son los básicos e indispensables para ejecutar un proceso de Verificación y Validación.
  - Algunos modelos no cuentan con expertos o consultores a nivel nacional que brinden la asesoría requerida.
  - No cuentan con plantillas o guías detalladas y prácticas que permitan llevar a cabo las actividades principales del proceso de Verificación y Validación.
- Por otro lado, debido a las presiones de tiempo y presupuesto inherente a los proyectos, los miembros del equipo tienden a abandonar o pierden el interés por las actividades de Verificación y Validación, el resultado final: un producto que no cumple las expectativas de los clientes o sobrepasan el costo y tiempo inicialmente planificado.

Algunas ventajas, aportes e impactos positivos que aporta este estudio son:

- Marco Metodológico del proceso de Verificación y Validación de Software para las Pymes que se dedican al desarrollo de Software que será fácil de implementar: se incluye la descripción de los principales procesos, roles/perfiles básicos que se deben adquirir para ejecutar los procesos requeridos, se incluirán también métricas y plantillas que puedan servir de guía para la implementación.
- Las Pymes que se dedican al desarrollo de Software podrán ejecutar el proceso de Verificación y Validación en sus proyectos y le permitirá entregar productos de mejor calidad.
- Las Pymes que se dedican al desarrollo de Software tendrán un impacto económico positivo. Reducirá el riesgo de ejecutar continuas re planificaciones, ampliaciones de presupuestos innecesarios, invertir horas extras para re-trabajos y altos niveles de rotación de personal ocasionados por el sobre esfuerzo de los recursos.



## 1.5 Alcance de la Investigación

- El alcance de este trabajo de investigación es desarrollar un Marco Metodológico para el proceso de Verificación y Validación de Software de una Pyme que se dedica al desarrollo de software basado en los modelos y estándares de calidad más representativos del mercado. Asimismo, se incluirá una guía metodológica detallada, roles, competencias del equipo de trabajo, las principales métricas, plantillas y lista de verificación base que deberían ser considerados durante la implementación de un Software de Calidad.
- Se seleccionó el proyecto de una Pyme o pequeña casa de desarrollo de Software, se aplicó el Marco Metodológico propuesto y se revisaron los resultados obtenidos. Para el presente trabajo de investigación se eligió el Sistema de Control de Inventarios de la empresa QBOSS como caso de estudio.

## 1.6 Organización de la Tesis

El presente trabajo está dividido en cinco capítulos, en el **Capítulo I** se muestra los antecedentes del problema, la definición del problema, los objetivos de la investigación, la motivación y alcance de la investigación, en el **Capítulo II** se presenta el Marco Teórico, en el **Capítulo III** se muestra Estado del Arte, se hace una revisión de los principales modelos, estándares, estudios sobre el proceso de Verificación y Validación de Software, en el **Capítulo IV** se muestra el Marco Metodológico propuesto para el Proceso de Verificación y Validación de Software, se indican los pasos que se deben seguir para su implementación, el perfil/competencias esenciales que debe tener los recursos, se detallan las métricas principales y las plantillas/listas de verificación que sirven como base para la Verificación y Validación de cada proceso. En el **Capítulo V** se aplicó el Marco Metodológico propuesto a una Pyme y se muestran los resultados obtenidos. Finalmente, en el **Capítulo VI** se muestran las conclusiones y trabajos futuros.



## Capítulo 2: Marco Teórico

En el presente capítulo se definen los principales conceptos relacionados con el proceso de Verificación y Validación. Primero, se describe las cinco etapas de la evolución de este proceso, después se describe la diferencia entre Aseguramiento de Calidad y el proceso de Verificación y Validación de Software, se muestran las técnicas de Verificación y Validación estática y dinámica y finalmente se detallan los estándares existentes.

### 2.1 Evolución del proceso de Verificación y Validación

A continuación se repasará brevemente la evolución del proceso de Verificación y Validación:

#### a) Etapa 1- Hasta 1956 – La orientación de las pruebas a la solución de errores.

El primer artículo basado en las pruebas del software fue escrito por Alan Turing. En este artículo se discutía el uso de afirmaciones que hoy en día se podrían denominarse como “pruebas de corrección”. Se introdujeron conceptos orientados a comprobar si un programa demostraba inteligencia.

Conceptos donde se establecía que un programa era la realización de los requisitos y la necesidad de saber si el resultado final cumplía con estos. Turing definió el desarrollo de una prueba con el fin de demostrar el comportamiento inteligente de una aplicación. Para llevar a cabo dicha prueba, Turing estableció que el funcionamiento de una aplicación y de un sistema de referencia, en este caso el ser humano, deberían resultar indistinguibles para un tercero, el cual asumiría el papel del probador (TURING. A., 1950).

#### b) Etapa 2- 1957-1978 - La orientación de las pruebas a la demostración.

En 1957, Charles Baker distinguió el concepto de “depuración” de “prueba” en su revisión “Mathematical Tables and Other Aids to Computation” del libro

“Digital Computer Programming” de Dan MacCracken. En la cual se destacaba que la verificación de los programas perseguía dos objetivos fundamentales:

- Asegurar el funcionamiento de la aplicación.
- Asegurar que la aplicación resolvía el problema para el que había sido planteada.

Para ello, estableció el concepto de depuración en referencia al primer objetivo y el de prueba al segundo. Además, la confianza que se refleja a través de la expresión del término “asegurar”, resultó frecuentemente utilizado en la definición de los objetivos de las pruebas con el fin de demostrar que el software cumplía con sus requerimientos (BAKER. C., 1957).

Durante este periodo, se produjo un notable incremento de los desarrollos software en términos de cantidad, costo y complejidad. Lo que condujo a que las pruebas aumenten su importancia dentro de la industria debido a la necesidad de asumir grandes riesgos económicos, un mejor desarrollo y una mejora en el mantenimiento del software.

Por aquel entonces, tanto los gerentes como los propios usuarios empezaron a tomar consciencia de que la cantidad de deficiencias que presentaban los sistemas informáticos era excesiva, y que su solución y la recuperación de los problemas que ocasionaban resultaban demasiados costosos. Por lo cual, se comenzó a dar un mayor énfasis a la “mejora de las pruebas”, concepto que perseguía dar una mayor efectividad a las pruebas a la hora de detectar problemas antes de la entrega de los productos.

Durante estos años, los conceptos de depuración y de prueba marcaron el esfuerzo por detectar, localizar, identificar y corregir los fallos. Las diferencias entre estas dos actividades se basaban en la definición del éxito (la resolución de los problemas). Pero la aparición del modelo de detección unió dichas tareas de modo que todo fallo detectado era incluido en las “pruebas” y todo

fallo localizado, identificado y corregido resultaba “depurado”. Aunque la depuración y la prueba reajustaron sus significados, aún era patente la necesidad de hacer más seguras las ejecuciones de los programas.

En esta etapa, debido a que llevar a cabo las pruebas suponía un esfuerzo comprendido de entre un 30% y un 50% (BEIZER. B., 1990), como mínimo, del costo total de un desarrollo software. Autores como Ince (INCE.D, 1987) y Jessop (RAMAMOORTHY,C., Ho, S., AND CHEN, W., 1976) llegaron a la conclusión de que si el proceso de pruebas pudiese ser automático se reduciría significativamente su costo.

Por ello, debido a que el peso de la generación de los datos utilizados en la construcción de los casos de prueba del producto software se consideraba crucial para el éxito de la prueba, en base a la imposibilidad de obtener un programa software completamente probado (el número de casos de prueba necesarios para probar un programa software es infinito (MYERSG F., 1979)). Y que un diseño adecuado de los casos de prueba puede detectar un elevado número de fallos. A mediados de los años 70 distintos autores presentaron metodologías para la generación automática de datos de prueba orientados a la trayectoria en las publicaciones: “A formal system for testing and debugging programs by symbolic execution” (BOYER. R, ELPAS. B, LEVITT. K, 1975) de Boyer, “A system to generate test data and symbolically execute programs.” (CLARKE, 1976) de Clarke,” Automated test data generation” (KOREL B., 1990), de Korel y “On the automated generation of program test data” (RAMAMOORTHY,C., Ho, S., AND CHEN, W., 1976) de Ramamoorthy. Estableciendo el método orientado a la trayectoria como un proceso de selección de los posibles recorridos que se pueden realizar a la hora de alcanzar un estado, generando datos pruebas para recorrerlos y teniendo en cuenta que estos puedan ser creados de forma automática.

### **c) Etapa 3- 1979-1982 - La orientación de las pruebas a la detección.**

En 1979, (MYERSG F., 1979) Myers describió el modelo orientado a la detección de pruebas y definió éstas como “el proceso de ejecutar un programa con la intención de encontrar errores”. En esta definición, se estableció con el objetivo principal de la detección de fallos en la implementación, pero la preocupación de Myers estaba centrada en la selección de objetivos que establecían que un software no tenía fallos. Ya que uno mismo puede llegar seleccionar inconscientemente datos de prueba que tienen una baja probabilidad de ocasionar fallos en el programa. Por lo tanto, si el objetivo es demostrar que un programa falla, los datos de prueba deberían tener una alta probabilidad de detectarlos. El cambio en la orientación de las pruebas de la demostración a la detección trajo consigo la asociación de las pruebas con otras actividades de detección de fallos. Tanto en el libro de Myers como en otras publicaciones nacidas en este periodo: “Software Project Verification and Validation” (DEUTSCH, M.,1979) de Deutsch y “Software Testing and Validation Techniques” de Miller (MILLER. E. HOWDEN.W.E.,1981), se analizó el uso de las pruebas en el desarrollo del software y su empleo en las técnicas de revisión. Otros artículos como los de “Software Project Verification and Validation” de” (DEUTSCH, M.,1979) y “Life-Cycle software Validation” de Howden (HOWDEN W.E., 1978) contienen las primeras descripciones de las aproximaciones a la detección combinada de fallos.

En 1982, surge una nueva metodología orientada a la automatización de casos de prueba denominada “aleatoria”. Publicada en “Automatic generation of random selfchecking test” (BIRD.D and Muñoz.D, 1982) por Bird y Muñoz, resultando ésta una técnica muy simple que crea datos aleatorios para cualquier caso que se quiera probar, lo cual presenta un bajo índice de cobertura.

#### **d) Etapa 4- 1983-1987 - La orientación de las pruebas a la evaluación.**

En 1983, el “Institute for Computer Sciences and Technology of the National Bureau of Standards” publica una línea de investigación [NBS] especialmente orientada a los sistemas de procesamiento de información federal (FIPS). En la cual se describe una metodología que integra el análisis, la revisión y las actividades de prueba con el fin de ofrecer una evaluación del producto a lo largo del ciclo de vida del software.

Estableciendo la necesidad de obtener en cada fase del ciclo de vida un conjunto de actividades y de productos asociados. Esta guía de desarrollo recomienda tres conjuntos de técnicas de evaluación; la básica, la comprensiva y la crítica. Dichas técnicas presentan un desarrollo evolutivo, de tal forma que cada una de ellas contiene lo desarrollado por su predecesora.

La filosofía colectiva de la cual se hace gala en la metodología FIPS, es fácilmente reconocible en la siguiente cita: “Ninguna técnica de verificación y validación puede garantizar la corrección, es decir, un software libre de errores. Sin embargo, una cuidadosa elección de técnicas para un proyecto específico puede ayudar a asegurar el desarrollo y el mantenimiento de la calidad del software de un proyecto” Pero FIPS no fue el único modelo que surgió en el periodo del 83-87. En este periodo fueron creados dos estándares relativos al desempeño de buenas prácticas a lo largo del desarrollo de las pruebas en la ingeniería del software. El primero de ellos, se encontraba relacionado con la documentación del proceso y el segundo hacía referencia a las pruebas unitarias.

En 1979, dado el desarrollo de las pruebas hasta ese momento, Glenford J. Myers (MYERSG F., 1979) promovió que la Ingeniería del Software separase las disciplinas fundamentales del desarrollo del software de la verificación y validación, disciplina que abarcaría las pruebas del software. El desarrollo de

esta disciplina se basó en las pruebas para dar lugar técnicas como las inspecciones y revisiones; y estrategias como las de caja negra y blanca. A pesar de esto, el concepto de verificación y validación no estaba claro.

Ambos términos resultaban habitualmente confundidos por lo que Boehm (BOEHM B., 1979) los matizó, estableciendo la verificación a través de la cuestión: “¿Se está construyendo el producto de la manera correcta?” y la validación a través de: “¿Se está construyendo el producto correcto?” con el fin de orientar sus respectivas actividades.

Ese mismo año, un grupo del Comité Técnico de Ingeniería del Software del IEEE comenzó a trabajar en un estándar para la documentación de las pruebas del software. Este proyecto no se encontraba orientado a estandarizar las mejores prácticas realizadas en aquel entonces, en él se buscó un consenso en la forma de realizar la documentación de las pruebas. Para lo cual, la documentación fue tomada como un sistema de datos estructurados que debería satisfacer la información y el acceso a los requerimientos de sus usuarios. Como resultado, el estándar ANSI/IEEE STD 829-1983 (ANSI/IEEE 829,1983) fue publicado en 1983 especificando el contenido y el formato de ocho documentos estándares. En los cuales se tuvo especial consideración con los aspectos relativos a la modularidad, cohesión, acoplamiento, usabilidad y facilidad de revisión de las pruebas.

Las principales diferencias existentes entre la propuesta ofrecida en el estándar ANSI/IEEE STD 829-1983 y las prácticas realizadas por aquel entonces, residían en las especificaciones de la planificación y el diseño de las pruebas. Un plan de pruebas según establece este estándar basa sus objetivos en función de la identificación de riesgos, el establecimiento de una estrategia general, la definición de una estructura para sus tareas, la distribución de los recursos y sus responsabilidades, el calendario de desarrollo y los posibles contratiempos que puedan surgir. Además, propone la identificación y la descripción de los casos específicos y los procedimientos de prueba con el fin



de diferenciar las especificaciones dadas. Pero muchos de los planes de pruebas realizados hasta entonces incluían tanto tareas de planificación como de diseño. Lo cual retrasaba considerablemente la planificación, limitando de esta forma la elección de estrategias.

A parte de diferenciar la especificación de casos y procedimientos, este estándar propone una especificación para el diseño de pruebas. Estableciendo una analogía entre las pruebas y la especificación de la arquitectura del software. Y centrando su propósito en la organización del conjunto de pruebas, de tal manera que se establece una relación directa de éstas con los requisitos del software. Lo cual abre una ventana a la adaptación de las pruebas al software, concepto que normalmente no se encontraba reflejado.

Un año más tarde, en 1984 el congreso del gobierno americano aprobó la creación de un organismo de investigación para el desarrollo de modelos de mejora para los problemas en el desarrollo de los sistemas de software, y evaluar la capacidad de respuesta y fiabilidad de las compañías que suministran software al Departamento de Defensa (DoD), dando lugar al Instituto de Ingeniería del Software (SEI). En él se comenzó un año más tarde a trabajar en un marco de madurez de procesos que permitiese evaluar a las empresas productoras de software.

Años más tarde, un segundo grupo perteneciente al IEEE comenzó a desarrollar un estándar basado en las pruebas unitarias. Como resultado, el estándar ANSI/IEEE STD 1008-1987 [tareas y documentos que IEEE88] fue publicado en 1987. En él se especifican las fases, actividades, comprenden las pruebas unitarias. Continuando la línea que establecía un fuerte paralelismo entre el desarrollo del software y el de las pruebas. Este estándar enfatiza la necesidad de diseñar un conjunto de pruebas, además establece la necesidad de realizar dos tipos de documentos a la hora de implementar las pruebas unitarias: una especificación del diseño de pruebas y un informe del resumen de las mismas. Estos requisitos fomentan el diseño de las pruebas y

representan su principal diferencia con las prácticas hasta el momento más comunes, ya que, normalmente, las pruebas de unitarias pasaban desapercibidas.

Un año después de comenzarse a desarrollar el estándar orientado a las pruebas unitarias, en 1985 sus autores introdujeron aquel proceso a lo largo de los distintos niveles de prueba existentes, dando como resultado una metodología conocida como “el proceso de evaluación y de pruebas sistemáticas” (STEP) (HETZEL, B.,1991). Esta Metodología definió un sistema de tareas de pruebas, productos y roles con el fin de dar consistencia y salvar costos a la hora de alcanzar los objetivos propuestos en las pruebas.

STEP se basa en un modelo de prevención del ciclo de vida paralelo al desarrollo software, estableciendo una secuencia de actividades como la planificación, el análisis, el diseño, la implementación, la ejecución y el mantenimiento de las pruebas.

#### **e) Etapa 4- 1988 - La orientación de las pruebas a la prevención.**

En 1989, Watts Humphrey y Ron Radice extendieron los principios desarrollados por Deming para su aplicación a la industria del software a través de sus trabajos en IBM y en el SEI.

Ya en la década de los 90, es publicado el libro “Software Testing Techniques” (BEIZER. B., 1990) el cual hace gala de un gran catálogo de técnicas de prueba. En él, Beizer comenta que “el acto de diseñar pruebas es una de las técnicas más efectivas a la hora de prevenir de defectos”, extendiendo así la definición de las pruebas como concepto de prevención de errores y promoviendo la realización de pruebas en las primeras fases del desarrollo. En ese momento, dado nivel de formalismo que poseían las pruebas, se detalla la definición del proceso de V&V en el IEEE/ANSI 610.12-1990 (IEEE/ANSI 610.12, 1990) estableciendo la verificación como “el proceso de evaluar un sistema o un

componente con el afán de determinar cuando los productos de una fase del desarrollo satisfacen las condiciones impuestas al comienzo de la misma.”; y la validación como “el proceso de evaluar un sistema o un componente durante o al final del proceso de desarrollo con el afán de determinar si satisface los requerimientos especificados”.

En ese mismo año, se introduce otra técnica basada en la automatización de los casos de prueba con la publicación de la metodología orientada al objetivo en “Automated test data generation” de Korel (KOREL B., 1990). La cual se establece como el proceso de generar datos de prueba con el fin alcanzar un determinado estado sin importar el camino que recorran estos.

En 1991, Hetzel (HETZEL, B.,1991) estableció el proceso de pruebas como la planificación, el diseño, la implementación y la ejecución de las pruebas y sus entornos. Incorporando todas las ideas que hasta la fecha habían considerado a las pruebas como un proceso gestionado, es decir, un ciclo de vida de vida implicado con las pruebas (BEIZER. B., 1990).

Pero 1991 fue un año especialmente significativo ya que también vio la luz el proyecto comenzado años más tarde por el SEI, el Modelo de Madurez de las Capacidades para el Software (CMMI-DEV, 2010). El cual, basado en el trabajo de Humphrey, constituye un marco de referencia de la capacidad de las organizaciones de desarrollo de software en el desempeño de sus diferentes procesos, proporcionando una base para la evaluación de la madurez de las mismas y una guía para implementar una estrategia para la mejora continua de los mismos.

## **2.2 Aseguramiento de Calidad de Software Vs Control de Calidad de Software**

### **2.2.1 Surgimiento de SQA (Software Quality Assurance)**

En los años 50 los proyectos del DoD (Department of Defense of USA) usualmente estaban muy alejados de la planificación, se pasaban del

presupuesto y tenían muchos problemas técnicos. Frecuentemente no funcionaban como se esperaba y muchos proyectos eran cancelados antes de ser entregados. Durante este periodo los contratistas para el desarrollo de software a menudo hacían estimaciones muy optimistas sobre el estado del desarrollo del software. El DoD normalmente no era notificado oportunamente de los problemas en la planificación, de la gestión del presupuesto y de los problemas técnicos. Para intentar resolver este problema se estableció la Verificación y Validación Independientes (IV&V del inglés Independent Verification and Validation), un proceso de ingeniería que empleaba metodologías rigurosas para evaluar la correctitud y calidad del software a lo largo de su ciclo de vida.

El primer software en usar IV&V fue el programa del misil atlas a finales de los años 50. Desde el proyecto atlas se ha recolectado mucha información que indica que los proyectos con IV&V se realizan o ejecutan mucho mejor que los proyectos sin IV&V. Con el tiempo el rol del IV&V se convirtió crítico. La actividad que llamamos SQA evoluciona directamente de la Verificación y Validación Independientes (IV&V), muchas de las tareas que asociamos con SQA son originarias de IV&V.

Luego durante los años 70 la actividad de desarrollo de software comenzó a expandirse y las compañías de desarrollo de software fueron experimentando los mismos pobres resultados que las agencias gubernamentales (DoD, NASA). Las compañías tenían dificultad para entregar el software dentro de los plazos, presupuesto y calidad planificados. Varios proyectos desarrollados entre 1980 y 1990 fueron desastrosos, muchos excedían ampliamente el presupuesto y la planificación o entregaban software de baja calidad que no se podía usar. Durante los 80 esta experiencia se convirtió en lo que conocemos como crisis del software, el tiempo consumido en el mantenimiento excedía el tiempo de la construcción de nuevos productos de software.

Luego de la crisis del Software en los años 80, SQA evoluciono hacia una herramienta que las compañías de desarrollo de software utilizaban para identificar de forma temprana los problemas de calidad en el proceso de desarrollo. Mientras SQA era visto como un pequeño paso dentro del proceso del desarrollo del software, muchos jefes de proyectos vieron beneficios cuantificables a partir de integrar SQA dentro del proceso de desarrollo de software.

En los 90 varias compañías de software ya tenían funciones de SQA dentro de sus organizaciones.

### **2.2.2 Definición de SQA (Software Quality Assurance)**

Al igual que ocurrió con la definición de calidad hay varios puntos de vista desde donde se puede definir el aseguramiento de la calidad del software. Desde el punto de vista de la evidencia, la IEEE define el aseguramiento de la calidad como “Una guía planificada y sistemática de todas las acciones necesarias para proveer la evidencia adecuada de que un producto cumple los requerimientos técnicos establecidos. Un conjunto de actividades diseñadas para evaluar el proceso por el cual un producto es desarrollado o construido.”

Daniel Galin define SQA como: “Un conjunto, sistemático y planificado, de acciones necesarias para proveer la evidencia adecuada de que el proceso de desarrollo o mantenimiento de un sistema de software cumple los requerimientos técnicos funcionales tan bien como los requerimientos gerenciales para cumplir la planificación y operar dentro del presupuesto confinado”.

Desde el punto de vista de la visibilidad, el SEI define SQA como: “El aseguramiento de la calidad del software provee claro control del proceso que está siendo usado por el proyecto y del producto que se está construyendo.”

Desde el punto de vista del aseguramiento, Don Reifer define SQA como “El aseguramiento de la calidad del software es el sistema de métodos y procedimientos usados para asegurar que el producto de software alcanza sus requerimientos. El sistema involucra la planificación, estimación y monitoreo de las actividades de desarrollo realizadas por otros.”

Desde el punto de vista de la capacidad de uso Schulmeyer y McManus definen SQA como: “Las actividades sistemáticas que proveen evidencia de la capacidad o disponibilidad de uso del producto de software total.”

### **2.2.3 SQA no es lo mismo que SQC (Software Quality Control)**

Generalmente cuando le preguntamos a un profesional de sistemas que es lo que entiende por aseguramiento de la calidad del software, inmediatamente comienza a hablar de testing, algunos de ellos incluyen a la validación y verificación y luego empiezan a hablar de revisiones, las cuales son sólo extensiones del testing. Es decir, a menudo hay una confusión entre SQA y el testing (el cual actualmente forma parte del área de control de calidad del software SQC).

Haciendo sólo testing y revisiones no aseguramos la calidad de los productos, sino aseguramos el cumplimiento de especificaciones tanto funcionales como técnicas. En el desarrollo de software la diferencia entre SQC y SQA no está clara y estos términos a menudo se confunden, SQA se encarga de controlar el cumplimiento del proceso, mientras que SQC son aquellas acciones del aseguramiento de la calidad que proporcionan un medio para controlar y medir las características de un elemento, proceso o facilidad respecto a los requisitos establecidos. La siguiente tabla 2-1 expone sintéticamente las diferencias entre control de calidad y aseguramiento de la calidad.

Control de calidad	Aseguramiento de la calidad
Detecta problemas en los productos de trabajo.	Asegura la adherencia a los procesos, estándares y planes.
Verifica que los productos de trabajo cumplan con los estándares de calidad	Evalúa que los procesos, planes y estándares utilizados en el proyecto cumplan con los
Revisa el contenido del producto	Revisa procesos

Tabla 2-1 Control de calidad vs. Aseguramiento de la calidad (Elaboración Propia)

En conclusión, el rol del SQA es auditar que los distintos equipos de la organización, inclusive el de SQC siguen los procedimientos, estándares y procesos establecidos. El equipo de SQA debería establecer métricas para medir la efectividad del proceso. Como complemento el rol de SQC es tomar una actitud activa de verificación y validación del resultado o salida del proceso implementado.

#### 2.2.4 Esquema Conceptual del proceso de Verificación y Validación

Los esfuerzos de Verificación y Validación tienen como objetivo asegurar que el software satisfaga los requisitos del usuario y la calidad esperada. Para diferenciar ambos términos, (BOEHM B., 1979) formuló las siguientes preguntas. Verificación: ¿Estamos construyendo el producto correctamente?, Validación: ¿Estamos construyendo el producto correcto?

El **proceso de verificación** determina si los productos software de una actividad (o fase) cumplen con los requisitos o condiciones que tienen impuestas por las actividades precedentes. Por motivos de efectividad en costo y rendimiento, la verificación debería integrarse lo antes posible en los procesos (tales como los de suministro, desarrollo, operación o mantenimiento) que la emplean. Estos procesos pueden incluir análisis, revisión y pruebas (ISO/IEC 12207:1995).

El **proceso de validación** determina si los requisitos y el sistema o producto software tal como se ha construido cumplen con su uso específico previsto. La validación puede llevarse a cabo en etapas tempranas y como parte del apoyo a la aceptación del producto (ISO/IEC 12207:1995).

Todos los productos generados en el proceso de desarrollo pueden estar sujetos a los procesos de Verificación y Validación. Dentro de este proceso se pueden utilizar estrategias de análisis estático (Verificación y Validación Estática) o dinámico / pruebas (Verificación y Validación Dinámica).

La **Prueba del Software** es una actividad realizada para evaluar la calidad del producto y mejorarla, identificando defectos y problemas. Se trata de *Verificación Dinámica* del comportamiento de un programa contra el *comportamiento esperado*, usando un conjunto finito de casos de prueba, seleccionados de manera adecuada desde el dominio infinito de ejecución. Verificación dinámica implica que se ejecuta el programa bajo prueba con un conjunto de datos de entrada. Comportamiento esperado implica que debe ser posible decidir si la salida de la ejecución del programa es aceptable o no. El comportamiento observado puede ser revisado contra las expectativas del usuario, una especificación, o el comportamiento anticipado por requisitos implícitos o expectativas razonables (SWEBOK, 2004).

En la figura 2-1 se muestra dónde se aplicaría el proceso de Verificación y Validación:



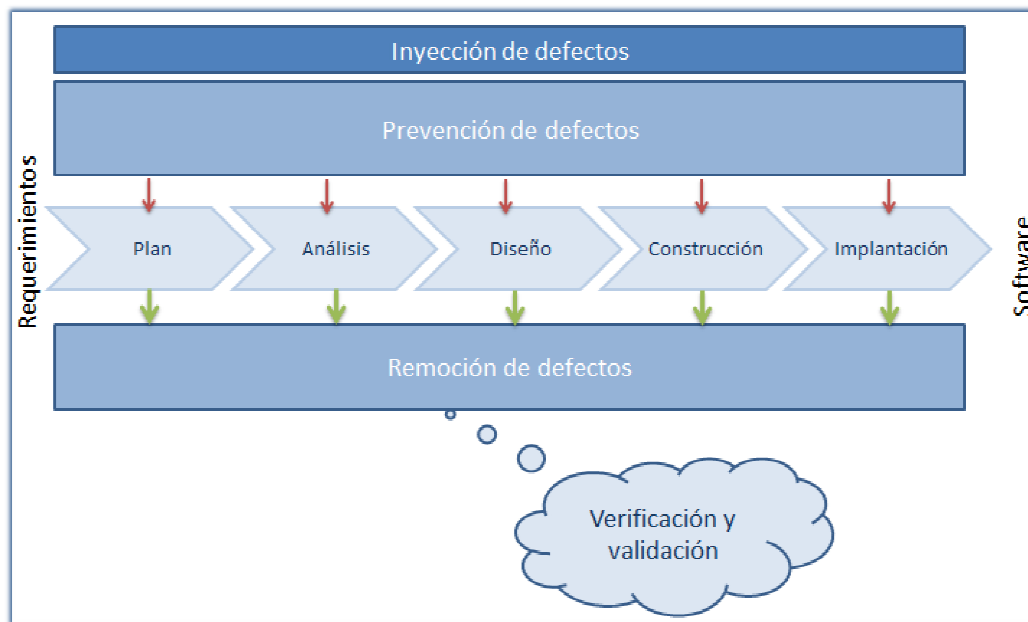


Figura 2-1. Aplicación del Proceso de V&V en el ciclo de vida del SW (ISO/IEC 12207:1995).

Como se puede apreciar en el gráfico durante cada fase del ciclo de vida se inyectan defectos, el proceso de Verificación y Validación debe ser aplicado durante todo el ciclo de vida desde la planificación hasta el proceso de implantación.

Durante y después del proceso de implementación, el software que se está construyendo debe ser comprobado para asegurar que satisface su especificación y entrega la funcionalidad esperada (SOMMERVILLE, IAN, 2005), el objetivo es detectar y remover los defectos encontrados antes de continuar con la implementación del Software.

### 2.3 Modelo V de Verificación y Validación

En la Figura 2-2 se presenta el modelo V que muestra cómo se relacionan las actividades de prueba con aquellas del desarrollo de software. El modelo V sugiere que la prueba unitaria y de integración se realice para verificar el código y también el diseño, comprobando que todos los aspectos del diseño detallado se han implementado correctamente en el código. La prueba del sistema debe verificar el diseño del sistema, asegurando que los aspectos relacionados con las cualidades del sistema están correctamente implementados. La prueba de

aceptación que tiene participación del cliente permite validar los requisitos. La vinculación entre la parte izquierda y derecha del modelo V implica que si se encuentran problemas durante la verificación y la validación entonces el lado izquierdo de la V puede ser ejecutado nuevamente para solucionar el problema y mejorar los requisitos, el diseño y el código antes de retomar las pruebas del lado derecho (PFLEEGER S.,2001; PRESSMAN, ROGER S., 2001).

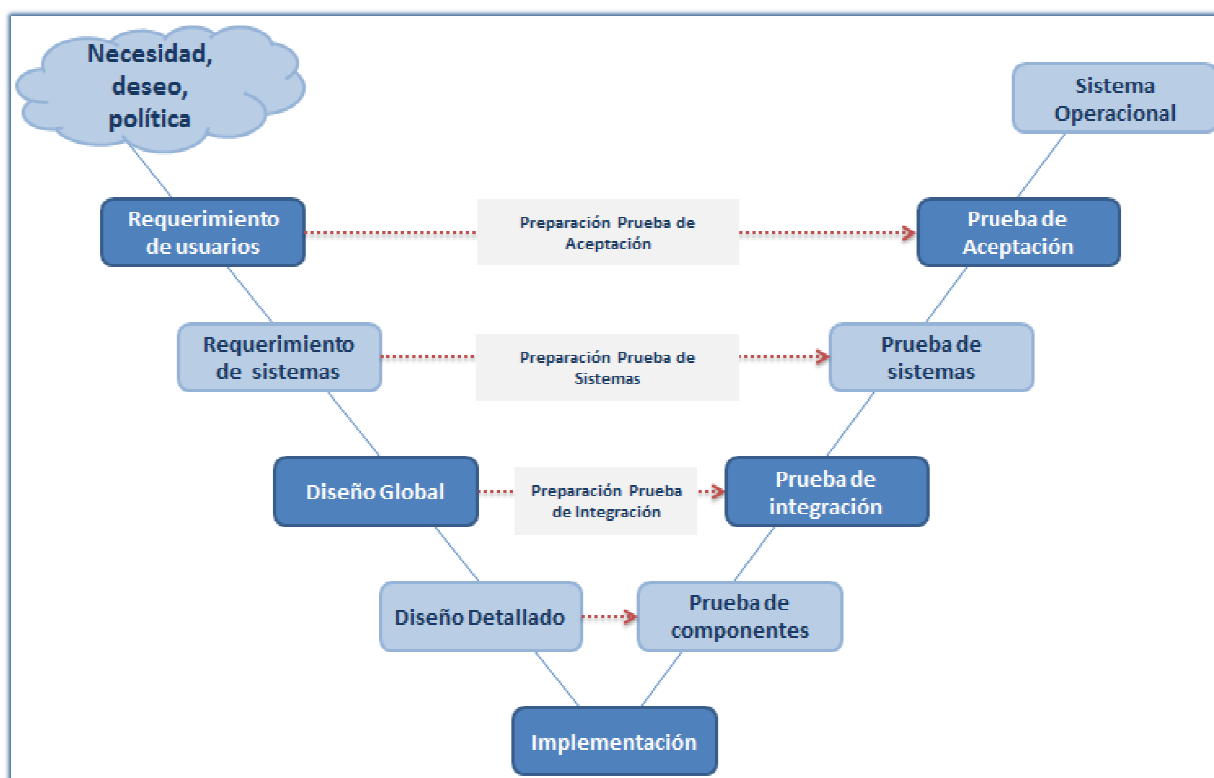


Figura 2-2 Modelo V del Proceso V&V (Pfleeger, 2001; Pressman, 2000).

A continuación, se presentan en forma resumida algunas de las estrategias dinámicas y estáticas para V&V y Pruebas del software. Entre las estrategias de análisis estático para V&V se resumen aquellas presentadas en la Tabla 2-2.

<b>Inspección</b> (IEEE 610.12-1990,1990; Pressman, 2000)	
<b>Objetivo:</b> examen visual de los productos del desarrollo para detectar errores (en el código, en la documentación de los requisitos, análisis y diseño), violaciones a los estándares u otros problemas.	<b>Aplicabilidad:</b> en todas las etapas del proceso de desarrollo.
<b>Walk-Through</b> (IEEE 610.12-1990,1990; Pressman, 2000)	

<b>Objetivo:</b> guiar a los participantes a través de una pieza de código o documentación, para que surjan preguntas y comentarios sobre posibles errores, violaciones a los estándares de desarrollo y otros problemas.	<b>Aplicabilidad:</b> en todas las etapas del proceso de desarrollo.
<b>Listas de Comprobación (listas de verificación)</b> (Kotonya, 1998)	
<b>Objetivo:</b> sirven de guía para la revisión, para identificar problemas en los requisitos en base a criterios de calidad esperados, como la facilidad para entender los requisitos, no redundancia, completitud, no ambigüedad, consistencia, conformidad con estándares y trazabilidad.	<b>Aplicabilidad:</b> en la etapa de requisitos.
<b>Escenarios</b> (IEEE 610.12-1990,1990)	
<b>Objetivo:</b> para validar los requisitos con el usuario. Los escenarios pueden describir casos relacionados con las funciones esperadas del sistema, como de nivel tecnológico. Se utilizan en reuniones de revisión con el usuario.	<b>Aplicabilidad:</b> en la etapa de requisitos.
<b>Prototipos</b> (IEEE 610.12-1990,1990)	
<b>Objetivo:</b> para validar los requisitos con el usuario. Un prototipo constituye una instancia o forma preliminar del sistema que sirve como modelo para las siguientes etapas y en esta fase permite validar que el sistema incluya las funcionalidades requeridas.	<b>Aplicabilidad:</b> en la etapa de requisitos.

Tabla 2-2 Estrategias de Análisis Estático (IEEE 610.12-1990; Pressman, 2000)

Las estrategias de análisis dinámico o pruebas del software se clasifican de acuerdo al Modelo V de la Figura 2-2. La prueba *estructural* es también llamada de caja blanca o basada en el código, y su objetivo es seleccionar los caminos de un programa para asegurar que sea perfecto en todos sus términos, aunque podría sin embargo no servir a la función que se pretende. Las pruebas estructurales nos convencerán de que un programa hace bien lo que hace; pero no de que haga lo que necesitamos. La prueba *funcional* es también llamada de caja negra o basada en la especificación, y su objetivo es validar desde el punto de vista *del* usuario si el comportamiento observado del software cumple o no con sus especificaciones, aun así un programa podría tener defectos internos que surgen en el momento más inoportuno. Las pruebas funcionales nos convencerán de que un programa hace lo que queremos pero no de que haga (además) otras cosas menos

aceptables (BEIZER. B., 1990; PRESSMAN, ROGER S., 2001). A continuación en la Tabla 2-3 se describe las estrategias de análisis dinámico:

<b>Prueba Unitaria (Ghezzi, 1991; Beizer, 1990; Binder, 2000; Pressman, 2000)</b>	
<b>Objetivo:</b> Identificar defectos relacionados a la lógica e implementación de cada una de las diferentes unidades individuales que constituyen un sistema. Se aplican estrategias estructurales y funcionales.	<b>Estrategias:</b> Estructural, basado en criterios de cobertura: sentencias, decisiones, condiciones, bucles, caminos, etc.; funcional, análisis de valores límites, partición en clases de equivalencia, tablas de decisión y grafos causa-efecto, etc.
<b>Prueba de Integración (Ghezzi, 1991; Beizer, 1990; Pressman, 2000)</b>	
<b>Objetivo:</b> detectar fallas de interacción entre las distintas unidades que componen al sistema. Permite asegurar que la funcionalidad conjunta de las unidades ya probadas individualmente, tengan un funcionamiento correcto. Los errores que se pueden identificar son: de interpretación, de llamada errónea, de interfaz.	<b>Estrategias:</b> en general se realiza una integración progresiva del sistema, descendente (top-down), ascendente (bottom-up) o una combinación de ambas. Se generan unidades auxiliares denominados drivers y stubs.
<b>Prueba Funcional (Beizer, 1990; Kaner et al., 1999; Myer, 2004; Bach, 2001,2003; Pérez et al., 2007; Black, 2002)</b>	
<b>Objetivo:</b> mostrar errores y discrepancias contra la especificación, en un software ya integrado y probado. Permite validar el comportamiento observado, pero no demostrar que el programa cumple su especificación.	<b>Estrategias:</b> pruebas de caja negra: análisis de valores límites, partición en clases de equivalencia, tablas de decisión, grafos causa-efecto, etc. Testing exploratorio, definido como el aprendizaje, el diseño y la ejecución de las pruebas en forma simultánea, aplicado cuando no es obvio cuál es la próxima prueba a realizar.
<b>Prueba de Sistema (Beizer, 1990; Swebok, 2004; Meyer, 2004; Meier et al, 2007; Ramakrishnan et al, 2001; Scambray et al, 2000)</b>	
<b>Objetivo:</b> estudiar el comportamiento del sistema en su totalidad, por medio de un	<b>Estrategias:</b> pruebas de volumen, de rendimiento (performance), de

conjunto de tipos de pruebas, que permiten verificar el cumplimiento de los requisitos no funcionales.	configuración, de compatibilidad, de recuperación, de seguridad.
<b>Prueba de Aceptación (Pressman, 2000; Kaner et al, 2001; Myer, 2004; SWEBOK, 2004)</b>	
<b>Objetivo:</b> comparar el programa contra sus requisitos iniciales y las necesidades reales de los usuarios. Es realizado generalmente por el cliente o el usuario final y se selecciona un subconjunto de los casos de prueba del sistema.	<b>Estrategias:</b> pruebas de usuario, realizada por el tipo de persona que usará el producto; prueba alfa, realizado por usuarios representativos en el entorno de desarrollo; prueba beta, realizado por usuarios finales en el entorno del usuario.

Tabla 2-3 Estrategias de Análisis Dinámico (Beizer, 1990; Kaner, 1999; Myer, 2004; Bach, 2001,2003; Pérez, 2007; Black, 2002)

## 2.4 Estándares de Verificación y Validación

En la siguiente tabla 2-4 se muestra los grupos y estándares más relevantes para el área de Calidad de Software.

Grupos Normas/ Estándares más relevantes del grupo	
<b>Normas ISO</b>	<p><b>ISO 9000 (2006):</b> define el vocabulario común a todos los documentos de la familia ISO 9000, aplicables a cualquier área de trabajo que requiera un sistema de gestión de la calidad</p> <p><b>ISO 9001 (2000):</b> toma el vocabulario del ISO 9000 y lo utiliza en la definición del marco conceptual de trabajo y requerimientos de un sistema de gestión de la calidad.</p>
<b>Normas ISO/IEC</b>	<p><b>ISO/IEC 9003:</b> en ella, el esfuerzo es invertido en hallar una correspondencia entre los requerimientos planteados en ISO 9001, las prácticas técnicas y de gerencia de desarrollo de software, esto con el objetivo de hallar una adaptación del modelo de sistema de gestión de la calidad aplicable al ámbito del software.</p> <p><b>ISO/IEC 9126:</b> se trata de una serie de cuatro partes que define un modelo para la calidad de un producto de software y sus métricas relacionadas.</p> <p><b>ISO/IEC 2500n-ISO/IEC 2504N (SQuaRE):</b> es una colección de 23 estándares y reportes técnicos reunidos bajo el nombre común de Software engineering-Software product Quality Requirements and</p>

	<p>Evaluation (SQuaRE). Estos documentos pueden clasificarse en cinco divisiones: Gestión de la Calidad, Modelo de Calidad, Medición de la Calidad, Requerimientos de la Calidad y Evaluación de la Calidad. SQuaRE reemplaza a la serie de normas ISO/IEC 9126.</p>
<b>Estándares IEEE</b>	<p><b>IEEE Std 730-2002:</b> provee los requerimientos formales mínimos para la realización de planes de aseguramiento de calidad de software para ser ejecutados en el momento en que el software es desarrollado o mantenido.</p> <p><b>IEEE Std 829-1998:</b> describe, en un grupo de sus cláusulas, ocho distintos documentos asociados con el proceso de pruebas de software, a saber: plan de pruebas, especificación del diseño de pruebas, especificación de casos de prueba, especificación de procedimientos de prueba, reporte de transmisión de ítems de prueba, registro de pruebas, reporte de incidente de pruebas y el reporte del resumen de pruebas. Los documentos son descritos como requerimientos.</p> <p><b>IEEE Std 1028-1997:</b> provee modelos para los cinco tipos distintos de revisión: revisiones gerenciales, revisiones técnicas, inspecciones, walk-throughs y auditorías. En cada modelo, el estándar identifica seis tipos de requerimientos: relativos a responsabilidades, a la entrada, a criterios de entrada, a procedimientos, a criterios de salida y a la salida.</p>

Tabla 2-4 Normas/Estándares Calidad (ISO 9001,2000; ISO 9000,2006; IEEE Std 730-2002; IEEE Std 1028-1997)

En la siguiente tabla 2-5 se muestra el estándar **IEEE 1012, 2004** del proceso de verificación y Validación y sus 6 subprocesos, cada uno con sus actividades y tareas definidas:

<b>V&amp;V y Revisiones</b>	<ol style="list-style-type: none"> <li>1. V&amp;V del Sistema y Generación del Plan de Pruebas.</li> <li>2. V&amp;V de Aceptación y Generación del Plan de Pruebas</li> <li>3. V&amp;V del Componente y Generación de Plan de Pruebas</li> <li>4. V&amp;V de la Integración y Generación del Plan de Pruebas</li> <li>5. V&amp;V del Componente y Generación del Diseño de Pruebas</li> <li>6. V&amp;V del Sistema y Generación del Diseño de Pruebas.</li> <li>7. V&amp;V de la Aceptación y Generación del Diseño de Pruebas</li> <li>8. V&amp;V del Componente y Generación de Casos de Pruebas.</li> <li>9. V&amp;V de la Integración y Generación de Casos de Prueba.</li> <li>10. V&amp;V de la Integración y Generación de Casos de Prueba.</li> <li>11. V&amp;V de la Aceptación y Generación de Casos de Prueba.</li> <li>12. V&amp;V del Componente y Generación del Procedimiento de Pruebas.</li> <li>13. V&amp;V de la Integración y Generación del Procedimiento de Pruebas.</li> <li>14. V&amp;V del Sistema y Generación del Procedimiento de Pruebas.</li> <li>15. V&amp;V del Componente y Ejecución de Pruebas.</li> <li>16. V&amp;V de la Aceptación y Generación del Procedimiento de Pruebas.</li> <li>17. V&amp;V de la Integración y Generación del Procedimiento del Pruebas.</li> <li>18. V&amp;V del Sistema y Ejecución de Pruebas.</li> </ol>
-----------------------------	--

	19. V&V de la Aceptación y Ejecución de Pruebas. 20. Auditoria de Configuración e Instalación. 21. Chequeo de Instalación 22. Generación del Reporte Final de V&V. 23. Revisión de SVVP.
<b>V&amp;V y Revisiones</b>	1. V&V del Sistema y Generación del Plan de Pruebas. 2. V&V de Aceptación y Generación del Plan de Pruebas 3. V&V del Componente y Generación de Plan de Pruebas 4. V&V de la Integración y Generación del Plan de Pruebas 5. V&V del Componente y Generación del Diseño de Pruebas 6. V&V del Sistema y Generación del Diseño de Pruebas. 7. V&V de la Aceptación y Generación del Diseño de Pruebas 8. V&V del Componente y Generación de Casos de Pruebas. 9. V&V de la Integración y Generación de Casos de Prueba. 10. V&V de la Integración y Generación de Casos de Prueba. 11. V&V de la Aceptación y Generación de Casos de Prueba. 12. V&V del Componente y Generación del Procedimiento de Pruebas. 13. V&V de la Integración y Generación del Procedimiento de Pruebas. 14. V&V del Sistema y Generación del Procedimiento de Pruebas. 15. V&V del Componente y Ejecución de Pruebas. 16. V&V de la Aceptación y Generación del Procedimiento de Pruebas. 17. V&V de la Integración y Generación del Procedimiento del Pruebas. 18. V&V del Sistema y Ejecución de Pruebas. 19. V&V de la Aceptación y Ejecución de Pruebas. 20. Auditoria de Configuración e Instalación. 21. Chequeo de Instalación 22. Generación del Reporte Final de V&V. Revisión de SVVP.

Tabla 2-5 IEEE 1012, 2014

### 2.4.1 ISO / IEC 29110

La ISO / IEC 29110 es un conjunto de normas e informes técnicos que se ha desarrollado para entidades muy pequeñas (VSE – Very Small Entities) .Una VSE se define como una entidad (empresas, organizaciones, departamentos o proyectos) que tiene menos de 25 personas.

El conjunto de documentos de la ISO/IEC 29110 ha sido desarrollado para mejorar la calidad de productos y/o calidad de los servicios y el desempeño de los procesos.

La ISO/IEC 29110-1 (ISO/IEC 29110) define los términos de negocio comunes al conjunto de documentos del perfil de la **PO** (Pequeñas Organizaciones).Introduce los conceptos de proceso, ciclo de vida y normalización. Así mismo presenta las características y requisitos de una **PO**

aclara los fundamentos para los perfiles, documentos, estándares y guías de una **PO** específica.

La ISO/IEC 29110-2 (ISO/IEC 29110) introduce los conceptos para el perfil normalizado de ingeniería de Software para las **PO** y define los términos comunes para el conjunto de documentos del perfil de las **PO**. Este especifica los elementos comunes para todos los perfiles normalizados (Estructura, conformidad, evaluación) e introduce la taxonomía (catálogo) de los perfiles de la ISO/IEC 29110 (ISO/IEC 29110).

ISO/IEC 29110-3 (ISO/IEC 29110) define los lineamientos y requisitos de conformidad de la evaluación de proceso, necesarios para alcanzar el propósito de los perfiles de la **PO** definidos. También contiene información que puede ser útil para desarrolladores de métodos y herramientas de evaluación. Está dirigido a personas que no tienen relación directa con el proceso de evaluación.

Para todas las organizaciones, pero en particular para las microempresas, las certificaciones internacionales pueden aumentar la credibilidad, la competitividad y el acceso a los mercados nacionales e internacionales.

Para las microempresas, un proceso de certificación debe ser simple, corto y de bajo costo, y tener credibilidad internacional. Brasil y Tailandia lideran el desarrollo procesos de certificación de ISO/IEC 29110 para cumplir con las necesidades del entorno micro empresarial.

Los perfiles se definen con el propósito de empaquetar referencias a y/o partes de otros documentos de manera formal, con el fin de adaptarlos a las necesidades y características de las VSE. Preparar un perfil implica producir dos tipos de documentos:



Marco de trabajo y taxonomía: Especifica los elementos comunes a todos los perfiles (estructura, conformidad, evaluación) e introduce la taxonomía (catálogo) de los perfiles ISO/IEC 29110.

Especificaciones de perfil: Proporciona la composición definitiva de un perfil, los enlaces normativos al subconjunto normativo de estándares usados en el perfil, y los enlaces informativos (referencias) a documentos de “entrada”.

Para cada perfil existe un documento de este tipo. Un ejemplo de una especificación de perfil es el documento (Especificación– Perfil Básico). Su objetivo es definir una guía de gestión de proyectos y desarrollo de software, adaptada a las necesidades de las VSE, para un subconjunto de procesos de ISO/IEC 12207 (ISO/IEC 12207:1995)

A continuación en la Figura 2-3 mostraremos la estructura de la ISO 29110:

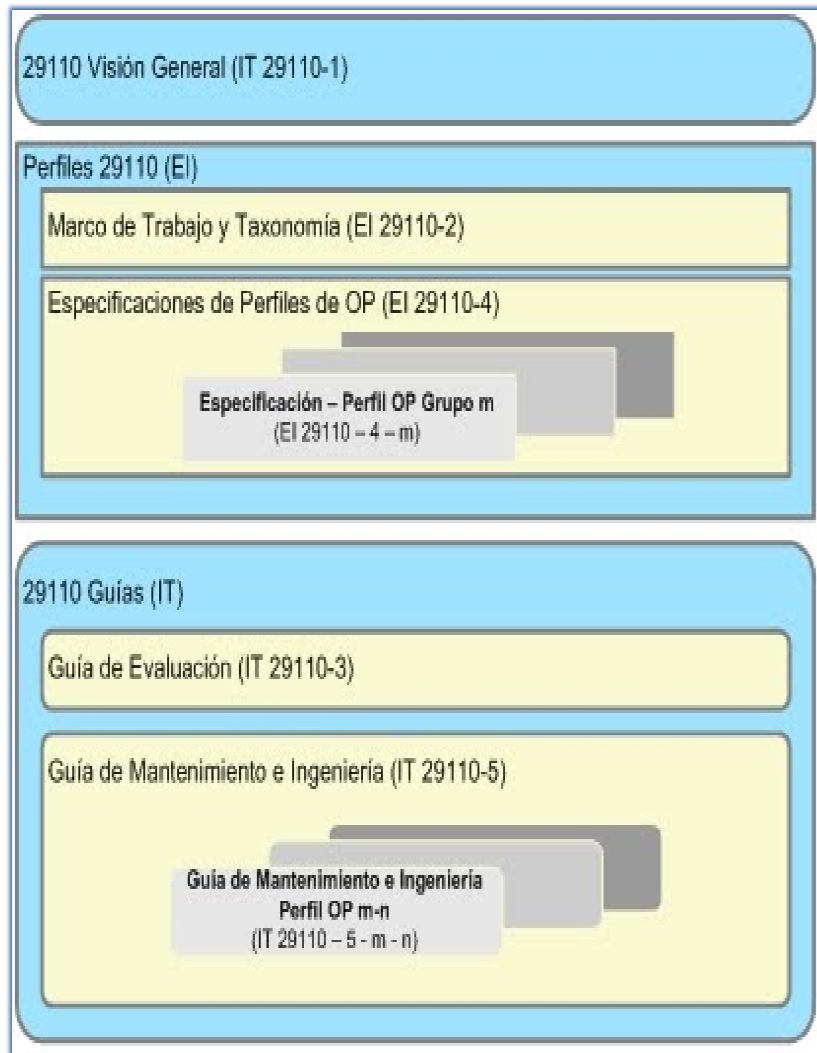


Figura 2-3 Estructura ISO/IEC 29110 (ISO/IEC 29110, 2013)

Esta norma puede ser la solución a multitud de pequeñas empresas, ya que el propósito del mismo es, como se ha expuesto anteriormente, hacer las normas actuales de ingeniería del software más accesibles a este tipo de empresas, adaptándolas a su tamaño y necesidades de negocio, lo cual es importante para su supervivencia en los mercados y para lograr competitividad, enfrentándose a problemas como la globalización creciente de los mercados y los cambios tecnológicos que se están produciendo

### **Verificación y Validación ISO/IEC 29110**

El estándar ISO/IEC 29110 define que las tareas de verificación y validación son necesarias para cada uno de los “Productos de Trabajo”. Las tareas de V&V

deben realizarse utilizando los criterios definidos en los procesos de desarrollo para así asegurar la coherencia entre los “Productos de Trabajo” de salida y entrada, definidos en cada una de las actividades. Los defectos son identificados, corregidos y los resultados son almacenados en los “Registros de Verificación/Validación”.

El estándar define las siguientes buenas prácticas para las actividades de V&V.

### **Proceso de Verificación de Software**

- a) una estrategia de verificación es implementada y desarrollada;
- b) se identifican los criterios de verificación para todos los productos de trabajo de software requeridos;
- c) las actividades de verificación son llevadas a cabo;
- d) los defectos son identificados y registrados; y
- e) los resultados de las actividades de verificación están a disposición de los clientes y otras partes interesadas.

### **Proceso de Validación de Software**

- a) una estrategia de validación es implementada y desarrollada;
- b) se identifican los criterios de validación para todos los productos de trabajo requeridos;
- c) las actividades de validación requeridas son llevadas a cabo;
- d) los defectos son identificados y registrados; y
- e) los resultados de las actividades de validación están a disposición de los clientes y otras partes interesadas.

A continuación en la Figura 2-4 se muestra el Diagrama del Proceso de Implementación de Software.

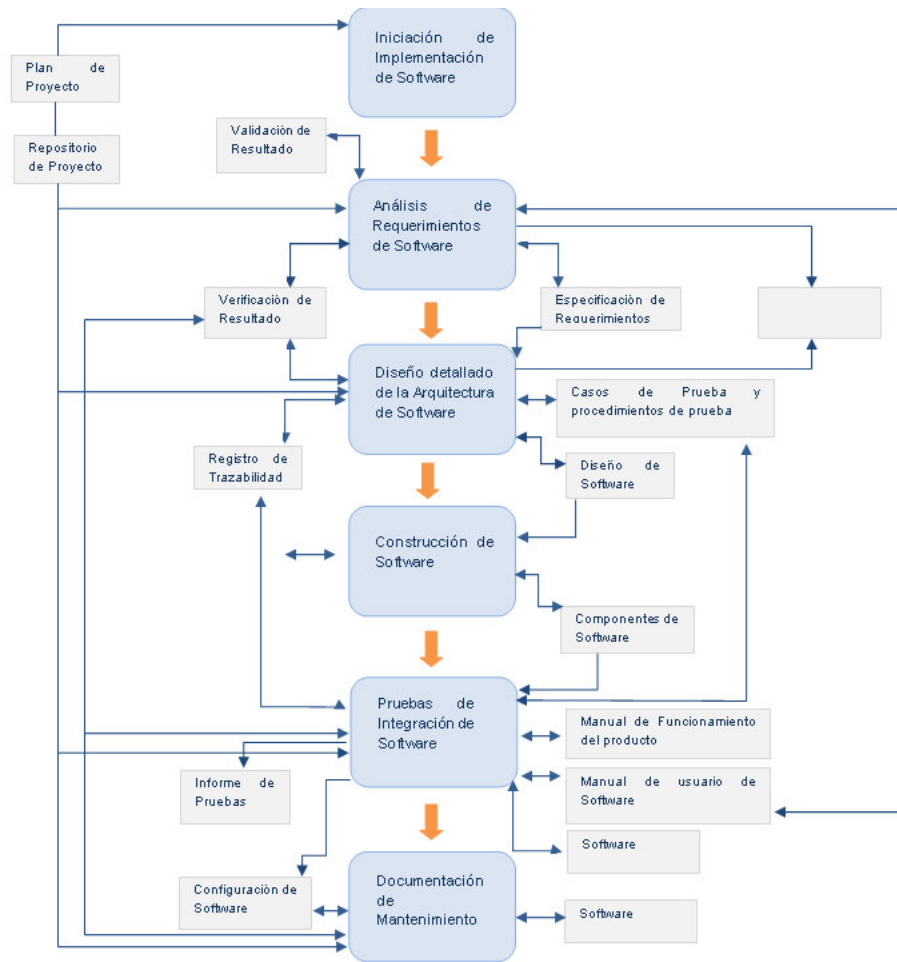


Figura 2-4- Diagrama del Proceso de Implementación de Software (ISO/IEC 29110, 2013)

En el presente Capítulo II se definieron los principales conceptos, técnicas y estándares que sirvieron como base teórica para entender el proceso de Verificación y Validación. En el siguiente Capítulo III se mostrarán los principales modelos relacionados con la calidad de software y el proceso de Verificación y Validación de software.

## Capítulo 3: Estado del Arte del proceso

### Verificación y Validación para Pymes

En el presente capítulo definiremos el estado del arte del Proceso de Verificación y Validación. Se describen los modelos de Calidad, Pruebas y su enfoque en el proceso de Verificación y Validación, se muestra un cuadro comparativo y finalmente se realiza una crítica entre los modelos.

#### 3.1 Modelos para el proceso de Verificación y Validación

A continuación se describen los modelos más representativos y usados actualmente.

##### 3.1.1 COMPETISOFT

Esta Metodología nos indica que el proceso de Desarrollo de Software se compone de uno o más ciclos de desarrollo. Cada ciclo está compuesto de las siguientes fases: Requisitos, Análisis, Diseño, Construcción, Integración, Pruebas y Cierre. En todas las fases existen actividades de V&V que se realizan para asegurar la calidad y corrección de los distintos productos que se generan a lo largo del ciclo de desarrollo.

Las actividades de pruebas, si bien tienen una fase específica donde ocurren, se realizan en el tiempo en paralelo con el resto de las fases. En la Figura 3-1 se muestra, como ejemplo, que las actividades de Diseño de los Casos de Prueba y su validación pueden ser realizadas en paralelo con el resto de las fases:

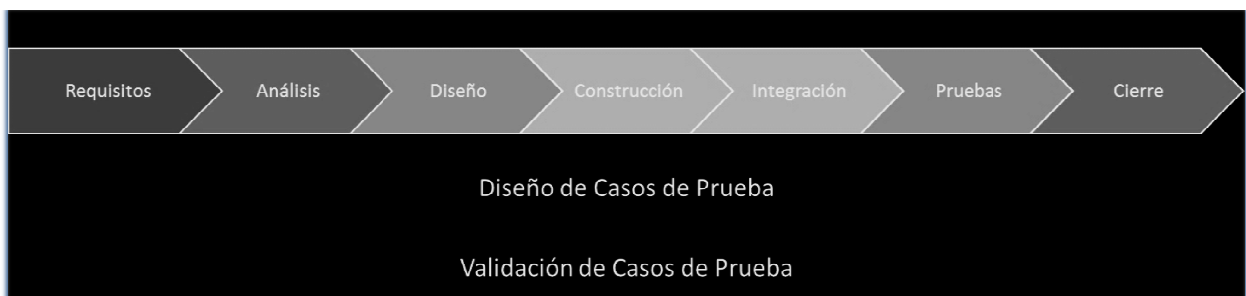


Figura 3-1 Actividades de Pruebas en el tiempo

Durante el desarrollo del producto, éste se construye en distintas iteraciones, por lo que el equipo de desarrollo genera distintas versiones del producto. Cada

versión puede agregar nuevas funcionalidades o mejorar las ya existentes en versiones previas. Así el producto va evolucionando hasta obtener una versión lo suficientemente madura para que sea liberada y entregada al cliente. Con cada versión del producto, se deben ejecutar las pruebas planificadas para esa versión.

En el caso de que el tiempo con el que se cuenta para la ejecución de las pruebas para una versión sea demasiado corto y no permita ejecutar todos los casos de prueba diseñados, se debe definir qué pruebas ejecutar para cada versión. Es conveniente utilizar un enfoque basado en los riesgos del producto (BACH, 1999). Un ejemplo de cómo planificar las pruebas funcionales para las distintas versiones de un producto basándose en los riesgos.

A continuación se describen las actividades de V&V en cada Fase de COMPETISOFT.

## **1.- FASE DE REQUISITOS**

Durante la fase de requisitos, las actividades de pruebas se focalizan en tres productos claves: Especificación de Requisitos, Plan de Pruebas y Manual de Usuario. A continuación se describen las actividades de validación y verificación para cada uno de estos productos.

### **▪ Verificar y Validar la Especificación de Requisitos**

Dentro de la fase de requisitos se deben levantar y documentar los requisitos del producto software. Esta actividad incluye generar la especificación de requisitos, la cual debe incluir los requisitos del software, restricciones del ambiente de negocio y requisitos de calidad. Como se mencionó anteriormente el costo de encontrar un error se incrementa conforme se avanza en el proceso de desarrollo, por lo que una vez terminada la especificación de requisitos, ésta debe ser sometida a procesos de verificación y validación.

- **Elaborar y Verificar el Plan de Pruebas**

En el plan de pruebas se incluye: alcance de las pruebas, calendario, ciclos de prueba planificados y las condiciones que deben cumplirse para comenzar cada ciclo, requisitos necesarios de hardware y software para ejecutar las pruebas, estrategia de pruebas que será utilizada durante el proyecto, técnicas usadas para el diseño de las pruebas, roles y sus responsabilidades, procedimientos a seguir para el reporte y seguimiento de incidentes, análisis de los **riesgos** del proyecto de prueba, clasificación de los defectos usada en el proyecto especificación de qué reportes y otros productos se generarán durante el proyecto. La plantilla Plan de Pruebas es una guía para realizar esta actividad.

Para la planificación de la agenda se utiliza la lista de funcionalidades a probar en cada ciclo. En función de esto se estima la duración de las actividades para las pruebas. Para la estimación de las actividades se utilizará la información de proyectos anteriores referentes a actividades similares. El plan de pruebas es realizado por el Responsable de Pruebas y debe ser validado por el Responsable de Desarrollo de Software y el Cliente.

- **Verificar el Manual de Usuario**

Una técnica adicional de V&V de los requisitos del usuario es elaborar en etapas tempranas del desarrollo la documentación de soporte. Específicamente el manual de usuario. Esta documentación debe revisarse buscando verificar consistencia con la Especificación de Requisitos y consistencia con el estándar de documentación requerido.

El responsable de esta actividad es el Revisor y para documentar los resultados de esta actividad se utiliza la Plantilla de Informe de Verificación. Estos defectos deber ser comunicados al responsable del documento para su corrección y posterior aprobación.

En la fase de requisitos se puede omitir las actividades de elaboración y verificación del manual de usuario. Sin embargo esta actividad deberá realizarse a más tardar en la fase de integración y pruebas.

## **2.- FASE DE ANÁLISIS**

- **Verificar la especificación del sistema**

El objetivo de esta actividad es verificar la claridad de la documentación de la especificación del sistema factibilidad y consistencia con la especificación de requisitos y consistencia con el estándar de documentación requerido. Así mismo se debe verificar que la matriz de trazabilidad contenga las relaciones adecuadas entre los requisitos y los elementos de la especificación del sistema.

El responsable de la ejecución de esta actividad es el rol de Revisor. Los defectos encontrados deben ser documentados en un informe de verificación y comunicados a los responsables para su corrección y aprobación.

- **Validar la especificación del sistema**

El objetivo de esta actividad es validar que la especificación del sistema cumple con las necesidades y expectativas acordadas con el cliente. Los resultados de esta actividad deben ser documentados en un informe de validación.

En esta actividad participan el responsable de pruebas, el cliente y el equipo de seguridad. La validación con el cliente puede ser omitida previo acuerdo, pero se debe validar con el equipo de seguridad.

- **Elaborar el Plan de Pruebas de Integración**

En el plan de pruebas de integración se incluye: calendario, orden de integración de los componentes o subsistemas, las pruebas que se



aplicarán para verificar la interacción entre los componentes, requisitos necesarios de hardware y software para ejecutar las pruebas, estrategia de integración que será utilizada, procedimientos a seguir para el reporte y seguimiento de incidentes, especificación de qué reportes y otros productos se generarán durante el proyecto.

El orden de integración de los componentes o subsistemas estará guiado por la parte arquitectónica del Análisis y Diseño, utilizando la Matriz de Trazabilidad, y considerando los aspectos de complejidad e importancia. Según ello se seleccionará la estrategia de integración a aplicar: top-down, bottom-up o híbrido.

Para la planificación de la agenda se utiliza la lista de funcionalidades a probar según la Matriz de Trazabilidad. En función de esto se estima la duración de las actividades de prueba unitarias. El plan de pruebas de integración es realizado por el Responsable de Pruebas y es validado por el Responsable de Desarrollo de Software.

### **3.- FASE DE DISEÑO**

- **Verificar la especificación del sistema**

El objetivo de esta actividad es verificar que en la arquitectura incluida en la especificación del sistema están representadas todas las unidades funcionales del sistema. Así mismo se debe verificar que la matriz de trazabilidad contenga las relaciones adecuadas entre los requisitos y los elementos de la especificación del sistema.

El responsable de la ejecución de esta actividad es el rol de Revisor. Los defectos encontrados deben ser documentados en un informe de verificación y comunicados a los responsables para su revisión y actualización.

- **Validar la especificación del sistema**

El objetivo de esta actividad es validar que la arquitectura de la especificación del sistema cumple con las necesidades especificadas en la etapa de análisis. Los resultados de esta actividad deben ser documentados en un informe de validación.

En esta actividad participan el Responsable de Administración del Proyecto y el Responsable del Desarrollo. Esta actividad puede ser omitida previo acuerdo con el cliente.

#### **4.- FASE DE CONSTRUCCIÓN**

- **Realizar pruebas unitarias**

Esta actividad puede dividirse en tres sub actividades principales:

- Definir y aplicar pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada de la Especificación del Sistema.
- Corregir los defectos encontrados hasta lograr pruebas unitarias exitosas (sin defectos).
- Actualizar la Matriz de Trazabilidad, incorporando los componentes contruidos o modificados.

- **Verificar la Matriz de Trazabilidad**

El Revisor verifica que la Matriz de Trazabilidad sea consistente con los elementos contruidos o modificados y en su vinculación con las relaciones establecidas previamente entre los requisitos y los elementos de la especificación del sistema.

Los defectos encontrados deben ser documentados en un Informe de Verificación y comunicados a los responsables para su corrección y aprobación.

## **5.- FASE DE INTEGRACIÓN**

Al efectuar la integración los módulos o componentes ya han sido previamente verificados mediante pruebas unitarias y están listos para su integración de acuerdo a la estrategia de integración seleccionada. Luego se procede a efectuar tareas de Pruebas de integración como parte de la siguiente actividad.

### **▪ Realizar Integración y Pruebas**

Esta actividad comprende las siguientes sub actividades:

- Aplicar las pruebas siguiendo el Plan de Pruebas de Integración, documentando los resultados en un Reporte de Pruebas de Integración.
- Reportar los defectos encontrados en el Sistema de Seguimiento de Defectos.
- Corregir los defectos encontrados hasta lograr una prueba de integración exitosa (pruebas de regresión).
- Actualizar la Matriz de Trazabilidad.

Esta actividad es realizada por el Programador y validada por el Responsable de Pruebas.

## **6.- FASE DE PRUEBAS**

### **▪ Diseñar los casos de prueba funcionales**

A partir de las funcionalidades planificadas para el ciclo desarrollan los casos de prueba usando técnicas de caja negra. Esta actividad incluye diseñar las pruebas e identificar los datos de prueba. Se presentan las técnicas para el diseño de casos de pruebas funcionales. La plantilla "Casos de Prueba" puede resultar una guía para esta actividad.

- **Diseñar los casos de prueba funcionales**

Las técnicas de diseño de pruebas no funcionales varían dependiendo del tipo de prueba y del alcance determinado en las mismas. Las técnicas a utilizarse pueden derivar de las utilizadas en pruebas funcionales. Para pruebas de configuración, por ejemplo, se podrían determinar casos de prueba según el escenario de ejecución o para pruebas de performance, se podrían seleccionar escenarios basados en análisis de riesgos.

- **Diseñar los casos de prueba de aceptación**

Los casos de prueba con los que se aceptará el producto construido por parte del cliente pueden ser diseñados por el equipo de prueba y ser validados por el cliente durante la construcción del producto. Luego, cuando se obtiene la versión final del producto, el cliente podría aceptarlo si las pruebas de aceptación definidas previamente son ejecutadas sin incidentes.

- **Verificar los casos de prueba del sistema**

El objetivo de esta actividad es verificar que los casos de prueba definidos son entendibles y completos. En el caso de seguir una plantilla para su definición, se revisa que el caso de prueba sea conforme con dicha plantilla. La salida de esta actividad es un Reporte de Validación donde se describen los incidentes encontrados en los casos de prueba.

- **Corregir los defectos encontrados en los Casos de Prueba del Sistema**

El equipo de prueba corrige los incidentes encontrados en los casos de prueba descritos en el Reporte de Verificación realizado por el Revisor.

- **Validar los Casos de Prueba del Sistema**

El Responsable de Pruebas valida los casos de prueba con el Cliente. El cliente valida que los casos de prueba y los datos con los que se probarán son relevantes para el negocio. Los cambios pedidos por el cliente son reportados en el Reporte de Validación.

- **Corregir los defectos encontrados en los Casos de Prueba del Sistema**

El equipo de pruebas modifica los casos de prueba validados por el cliente, según el Reporte de Validación, y solicita al cliente la aprobación de las correcciones.

- **Realizar pruebas en el ambiente de pruebas**

En esta actividad se ejecutan los casos de prueba definidos en la actividad Diseñar los casos de prueba del sistema. Para cada versión del producto que es liberada por el equipo de desarrollo para ser probada por el equipo de pruebas.

1. Se instala el producto en el ambiente de pruebas y se configuran los datos necesarios para las pruebas.
2. Se ejecutan los casos de prueba, esta actividad puede ser manual o automatizada. Para cada caso de prueba se ejecuta y se compara el resultado obtenido en el producto contra el resultado esperado en el caso de prueba.
3. En caso de que el resultado real no se corresponda con el esperado, se reportan los defectos encontrados en el Sistema de Seguimiento de Defectos.

Se ejecutan primero las pruebas funcionales y luego el resto de las pruebas del sistema que se hayan definido en el Plan de Pruebas.

- **Corregir los defectos encontrados**

El equipo de desarrollo realiza las correcciones en el producto de acuerdo al Sistema de Seguimiento de Defectos.

- **Verificar y Cerrar los defectos**

El equipo de pruebas verifica que las correcciones se realizaron y que los defectos pueden ser cerrados. Para esto, se vuelve a ejecutar el caso de prueba que encontró el defecto y también otros casos de prueba para verificar que el arreglo no introdujo efectos secundarios en el producto (pruebas de regresión). Se actualiza el estado de los defectos en el Sistema de Seguimiento de Defectos.

- **Realizar las Pruebas de Aceptación del Sistema**

Se ejecutan los casos de prueba de aceptación que fueron definidos junto con el cliente en la actividad Diseñar casos de prueba del sistema. Los casos de prueba de aceptación se ejecutan en el entorno definido por el cliente siguiendo el Plan de Pruebas del Sistema, documentando los resultados en un Reporte de Pruebas de Aceptación. Los defectos encontrados se reportan en el Sistema de Seguimiento de Defectos.

### **3.1.2 CMMI for Development**

CMMI for Development (Capability Maturity Model Integration for Development) (SEI, 2010) es un modelo de referencia que cubre el desarrollo y mantenimiento de las actividades aplicadas tanto a los productos como a los servicios.

CMMI surge a partir de la versión 2.0 de SW-CMM con objetivo de solucionar los problemas existentes con los distintos CMM. La misión inicial de este proyecto fue combinar los modelos SW-CMM, SECM e IPD-CMM. Su primera versión fue publicada en agosto del año 2000. Poseía un núcleo común a los diferentes modelos CMM y después una parte específica propia de cada modelo; dos años después, se publicó la versión 1.1. La Figura 3-2 muestra la estructura de CMMI.

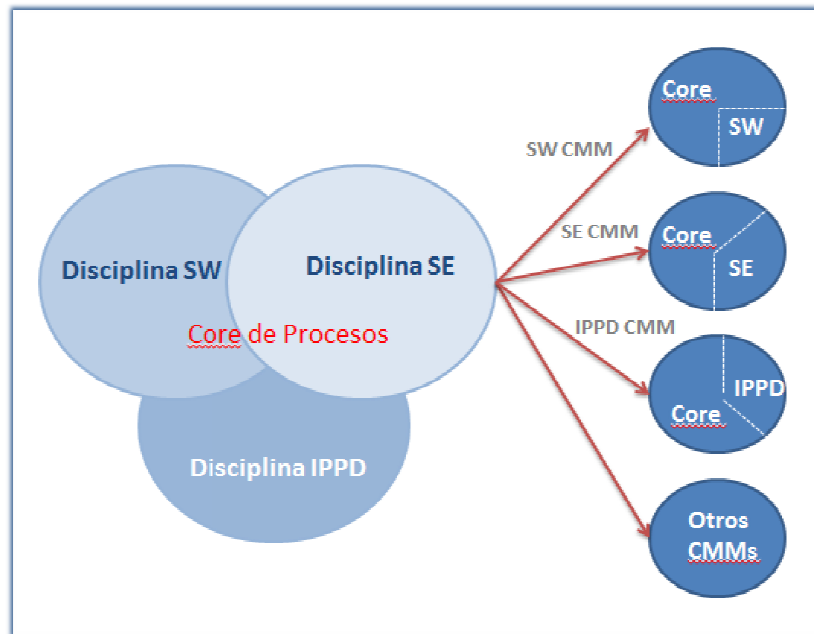


Figura 3-2 Estructura CMMI

Sin embargo, existía la necesidad de integrar a las diferentes disciplinas en un marco de trabajo combinado que permitiese dirigir tanto Software como Sistemas de Ingeniería o ambos, junto con IPPD. En Noviembre de 2010 se publica CMMI for Development v1.3, que es la versión más actual en lo que se refiere a buenas prácticas para el desarrollo de productos software. Ésta, además de ser una versión integrada al igual que la 1.1 y la 1.2, proporciona mejoras sobre la versión anterior, para solucionar los problemas que han ido surgiendo de la práctica. Es importante también reseñar que esta última versión del CMMI consta de otras dos extensiones, llamadas constelaciones, dirigidas a la mejora de los procesos de adquisición de productos y servicios (CMMI for Acquisitions) y a la mejora de los procesos de gestión del servicio (CMMI for services). Estos dos últimos no han sido estudiados, puesto que no incluyen buenas prácticas referentes a la validación y verificación de productos software.

CMMI define un conjunto de buenas prácticas que cubren un conjunto de áreas de proceso involucradas en el desarrollo y mantenimiento, permitiendo establecer tanto el nivel de madurez como el nivel de capacidad existente en la organización

(SEI, 2010). La Figura 3-3 muestra la estructura de cada una de las áreas de proceso de CMMI.

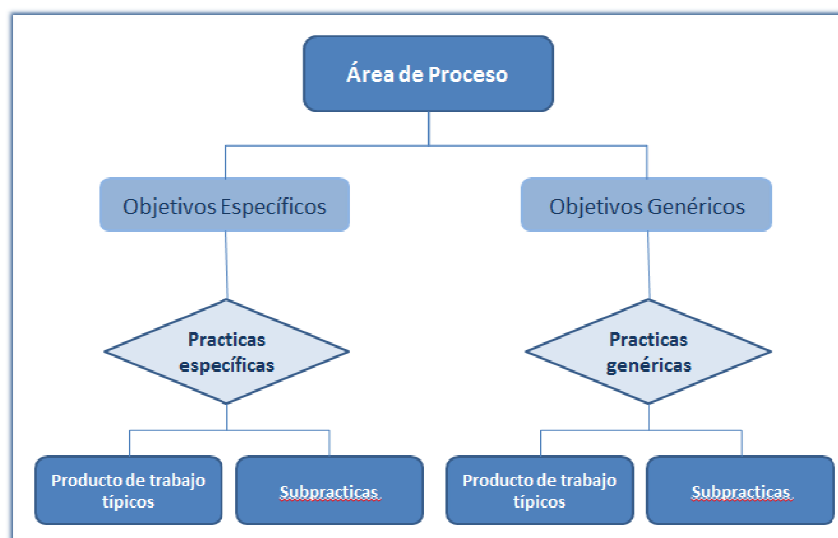


Figura 3-3 Estructura de Área de Proceso

También permite dos representaciones diferentes: continua y por etapas. La primera define 6 niveles de capacidad que permiten conocer la capacidad de los procesos que sean seleccionados para su mejora; se centra en un área de proceso y en adquirir un nivel de capacidad para el mismo. La segunda define cinco niveles de madurez: Inicial, Gestionado, Definido, Gestionado Cuantitativamente y Optimizado, que permiten conocer la madurez de los procesos que se realizan en la organización; centrándose en la madurez de toda la organización.

La Figura 3-4 muestra un ejemplo de la representación continua y la representación escalonada.





Figura 3-4 Representaciones Continua y Escalonada CMMI

En relación al proceso de pruebas define 3 áreas de proceso: Integración de Producto, Validación, y Verificación; sin embargo no cubren todas las necesidades del proceso de pruebas, puesto que no se encuentran definidas todas las mejores prácticas relacionadas con las pruebas software ni el conjunto de los productos de trabajo que son necesarios. Esto se debe a que, al igual que el resto de las áreas de proceso de CMMI, este estándar no entra al detalle de cómo deben de definirse e implementarse los procesos.

El propósito del área de Integración de Producto es llevar a cabo la integración de los distintos componentes que forman el producto, incluyendo las actividades necesarias para asegurar la compatibilidad de las distintas interfaces (SEI, 2010).

En relación al área de Validación, su propósito es demostrar que un producto o un componente de un producto satisfacen las necesidades para las cuales ha sido creado, dentro del entorno en el que deberá ser ejecutado. Para ello, CMMI propone dos actividades a llevar a cabo: Preparación para la validación y Validación de los productos o componentes (SEI, 2010).

Respecto al área de Verificación, su propósito es asegurar que un producto cumple con los requisitos que hayan sido especificados. Incluye la verificación tanto del producto final como de un producto de trabajo intermedio con respecto a los requisitos. Es un proceso incremental que ocurre a través del desarrollo del producto y productos de trabajo, comienza con la verificación de los requisitos, continua con la verificación de la evolución de los productos de trabajo, y finaliza con la verificación del producto completo. Para ello, CMMI propone llevar a cabo tres actividades: Preparación para la verificación, Desarrollo de revisiones por pares y Verificación de los productos de trabajo seleccionados.

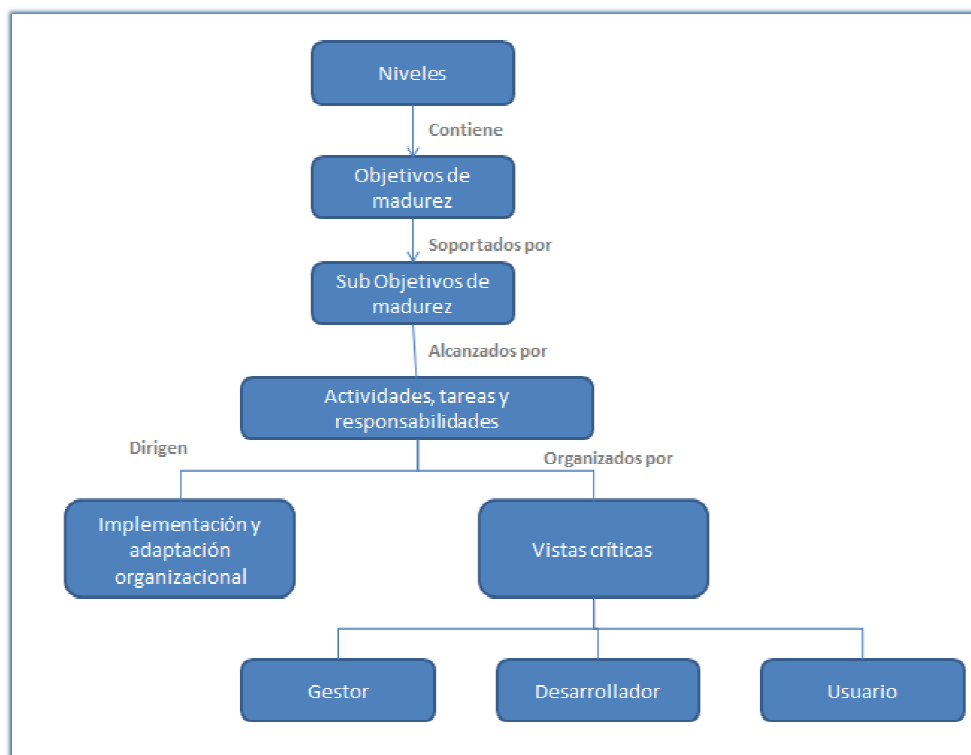
Como se ha visto, CMMI for Development 1.3 (SEI, 2010) es un estándar general para la mejora de los procesos de desarrollo de software, que define un conjunto de niveles y áreas de proceso a ser implementadas por las organizaciones para mejorar el desarrollo de productos software, considerando todas las fases que aparecen en el proceso. Sin embargo, este estándar es genérico y no define en detalle procesos ni prácticas a ser implementadas en ninguna de sus áreas de proceso, por lo que requiere del uso de otros estándares y de la experiencia del personal para definir los procesos adecuados dentro de la organización.

Además, dada su generalidad, no cubre el proceso de pruebas en toda su amplitud, simplemente define tres áreas de proceso que, como se ha visto, sólo incluyen algunas de las actividades básicas relacionadas con las pruebas. Por ello, este estándar no puede ser utilizado como modelo para la mejora del proceso de prueba en las organizaciones.

### **3.1.3 TMM - Test Maturity Model**

El modelo TMM-Test Maturity Model (BURNSTEIN I., 2003) ha sido desarrollado por el Instituto Tecnológico de Illinois como una guía para la mejora del proceso de pruebas. Al igual que CMM, modelo al que complementa, utiliza el concepto de niveles de madurez para la evaluación y mejora del proceso de pruebas; aunque a diferencia de su antecesor no contempla los niveles de capacidad.

TMM consta de 5 niveles de madurez que reflejan el grado de madurez de la organización en el proceso de pruebas; es decir, representan la evolución hacia un proceso de pruebas maduro. Cada nivel tiene definidas un conjunto de áreas de proceso, y cada una de ellas contiene las actividades relacionadas con la misma que es necesario realizar para contribuir a la mejora del proceso (BURNSTEIN, I., SUWANNASART, T., CARLSON C.R., 1996a) (BURNSTEIN, I., SUWANNASART, T., CARLSON C.R., 1996b). La Figura 3-5 muestra la estructura implementada en cada uno de los niveles de TMM.



*Figura 3-5 Estructura de Nivel TMM*

La Figura 3-6 muestra los niveles y objetivos de madurez de TMM. De acuerdo a (BURNSTEIN I., 2003), cada uno de los niveles tiene una serie de características.

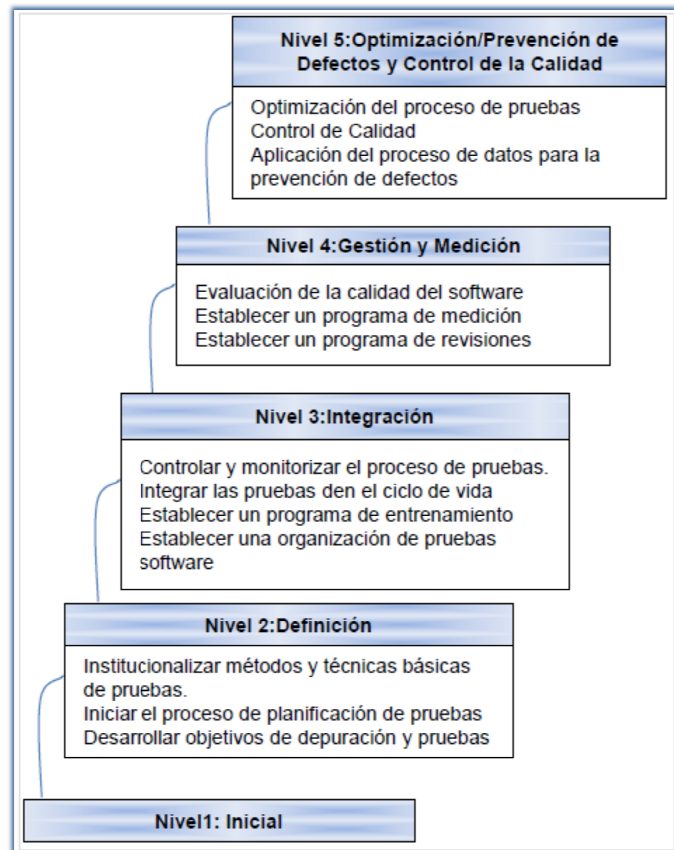


Figura 3-6 Niveles CMM

El nivel inicial se caracteriza porque el proceso de pruebas es un caos; no se encuentra definido y es considerado como una parte de la depuración. Las pruebas se desarrollan ad hoc después de haber terminado toda la codificación.

En el nivel 2, se separan las pruebas de la depuración, definiéndose una fase después de la codificación. El objetivo principal de este nivel es verificar que el software satisface los requisitos especificados. Sin embargo, surgirán problemas de calidad en este nivel, debido a que la planificación de las pruebas se realiza tarde. Además, los defectos se propagarán desde los requisitos y fases de diseño al código, ya que no existen programas de revisión.

En el nivel 3, las pruebas se encuentran integradas en todo el ciclo de vida del software, no son una fase que se realiza después de la codificación. A diferencia del nivel 2, la planificación de las pruebas comenzará con la fase de requisitos y

continuará durante todo el ciclo de vida. Existe una organización de pruebas, un programa de formación y se reconoce a las pruebas como una actividad profesional. Se realizarán revisiones aunque no exista un programa formal de revisión y éstas resulten inconsistentes.

El nivel 4 se caracteriza porque las pruebas son un proceso medido y cuantificado. Las revisiones de todas las fases del proceso de desarrollo son reconocidas como actividades de pruebas y control de calidad en este nivel. Se prueba el software atendiendo a los atributos de calidad, como fiabilidad, usabilidad y mantenimiento. Las deficiencias en este nivel se producen debido a la carencia en la prevención de defectos. Las actividades de prueba se expanden, incluyen revisiones, inspecciones y *walkthroughs* a través de las diferentes fases.

Finalmente, el nivel 5 se caracteriza porque el proceso de pruebas es repetible, definido, gestionado y medido, por lo que se pueden definir mecanismos que permitan una mejora continua en el proceso de pruebas. Se realiza prevención de defectos y control de calidad.

TMM podría ser utilizado en organizaciones que utilizan una estructura organizativa de pruebas basada en equipos independientes o grupos de SQA. En el caso de que los desarrolladores sean los *testers* o equipos integrados de pruebas este modelo de referencia resulta demasiado complejo y burocrático como para ser utilizado; tampoco es apropiado para una factoría de pruebas ya que no contempla procesos de gestión de servicios.

TMM presenta una serie de limitaciones a la hora de considerarlo como modelo de procesos, puesto que le sucede lo mismo que a CMMI. Esto es, define un conjunto de áreas de procesos pero no profundiza en la propia definición de los procesos y actividades a llevar a cabo para la validación y verificación de productos software, por lo que es la propia organización la que se tiene que ocupar de la definición de los mismos. Además, tampoco identifica las actividades que es necesario realizar

para “probar los productos”, lo cual es una tarea clave dentro de las pruebas de software.

Este modelo se ha quedado obsoleto con la desaparición de SW-CMM, por lo que en la actualidad ya apenas se usa. Sin embargo se ha desarrollado una evolución suya TMMi, consistente con las nuevas versiones de CMMI for Development.

### **3.1.4 TMMi - Test Maturity Model Integration**

El modelo TMMi - Test Maturity Model Integration (VEENENDAAL, E., 2010) ha sido desarrollado por la *TMMi Foundation* como una guía y un marco de referencia para la mejora del proceso de pruebas, siendo un modelo complementario a CMMI. Al igual que este último, TMMi define una representación por etapas y hace uso del concepto de niveles de madurez para evaluar y mejorar las diferentes áreas de proceso de cada uno de los niveles. Además de las áreas de proceso identifica el conjunto de objetivos a alcanzar y las prácticas que hay que realizar para ello. El objetivo de TMMi es soportar las actividades de prueba y la mejora del proceso de prueba tanto en las disciplinas de ingeniería de sistemas y de ingeniería de software. Al igual que CMMI, TMMi trata de proporcionar a las organizaciones la mejora de procesos, en este caso de procesos de prueba, para lo cual es sustancial contar con un mecanismo de evaluación que permita identificar las oportunidades de mejora. Para ello la *TMMi Foundation* ha elaborado el “TMMi Assessment Method Application Requirements”, que contiene los requisitos para la evaluación de TMMi, de forma que se puede conocer el nivel de madurez de una organización en lo referente al proceso de pruebas (GOSLIN, A. 2009).

TMMi consta de cinco niveles que van desde la realización de un proceso de pruebas de manera ad-hoc y no gestionada hasta optimizado pasando por gestionado, definido, y medido optimizado. Pasar de un nivel a otro supone haber alcanzado una mejora adecuado y sirve también como punto de partida para el siguiente nivel (VEENENDAAL, 2010). La Figura 3-7 se muestra la estructura de TMMi, es decir los niveles de madurez de que consta.

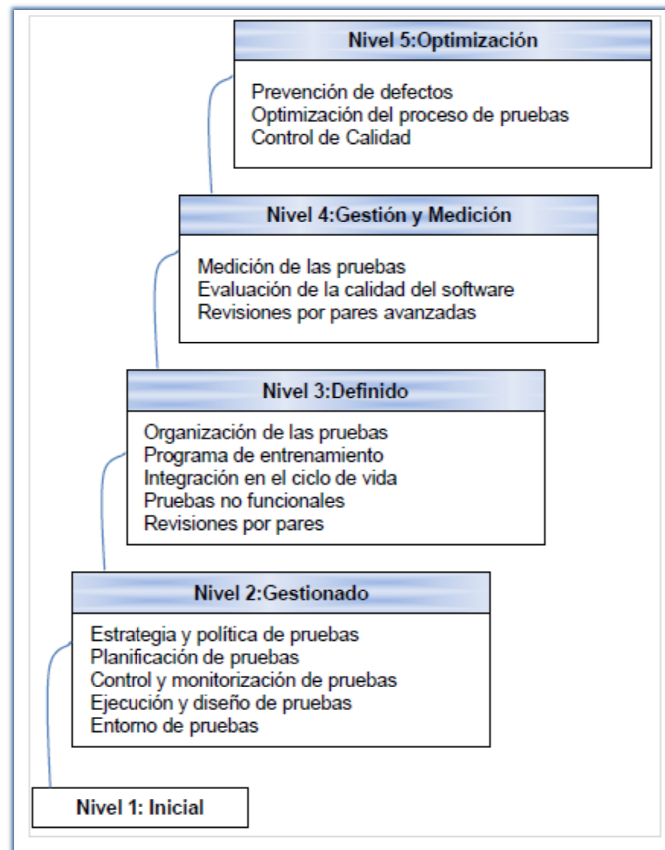


Figura 3-7 Niveles TMMI

El nivel 1 se caracteriza porque las pruebas se realizan de manera caótica, no hay un proceso definido y, frecuentemente, sólo son consideradas como parte de la depuración.

Las pruebas se realizan de manera ad-hoc una vez que la codificación se ha terminado, puesto que el objetivo de las pruebas a este nivel es comprobar si el software se ejecuta sin errores importantes. El éxito depende fundamentalmente de la competencia del personal a cargo de las mismas (VEENENDAAL, E., 2010).

En el nivel 2 se dispone de un proceso gestionado para la realización de las pruebas que está separado claramente de la depuración. El principal objetivo de este nivel es verificar que el producto satisface los requisitos establecidos

inicialmente. Consta de cinco áreas de proceso que son: Política y estrategia de pruebas, Planificación de pruebas, Monitorización y control de pruebas, Diseño y ejecución de pruebas y Entorno de prueba (VEENENDAAL, E., 2010).

A nivel 3 el proceso de pruebas se encuentra integrado dentro del ciclo de vida de desarrollo y cuenta con sus propios hitos. Las organizaciones que se encuentran en este nivel entienden la importancia de las revisiones y del control de calidad. Las áreas de proceso definidas para este nivel son: Organización de pruebas, Programa de formación en pruebas, Ciclo de vida e integración de pruebas, Pruebas no funcionales y revisiones por pares (VEENENDAAL, E., 2010).

Las organizaciones que se encuentran en el nivel 4 de TMMi disponen de un proceso de pruebas bien fundado, completamente definido y medible. Las pruebas se perciben como evaluación, por lo que las organizaciones ponen en práctica programas de mejorar del proceso de pruebas evaluación su calidad, la productividad y haciendo seguimiento de las mejoras. Este nivel consta de tres áreas de proceso: Medición de pruebas, Evaluación de la calidad del producto y revisión por pares avanzada (VEENENDAAL, E., 2010).

Finalmente, el nivel 5 se caracteriza porque la organización es capaz de mejorar continuamente su proceso de pruebas en base a un proceso controlado estadísticamente.

Los métodos y técnicas de pruebas son optimizados y existe un foco continuo en la mejora del proceso. Para alcanzar este objetivo se dispone de tres áreas de proceso: Prevención de defectos, Control de Calidad y Optimización del proceso de prueba (VEENENDAAL, E., 2010).

Como se ha visto, cada uno de estos cinco niveles consta de diferentes áreas de proceso dirigidas a que las organizaciones alcancen los objetivos establecidos para cada nivel. La figura 3-8 muestra la estructura de las áreas de proceso definidas en TMMi.



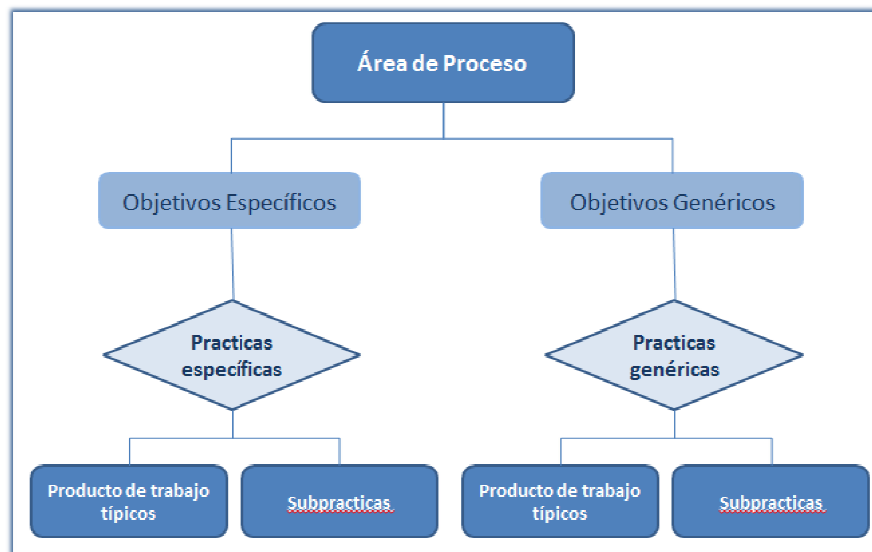


Figura 3-8 Estructura Área de Proceso TMMi

TMMi podría ser utilizado en organizaciones que utilizan una estructura organizativa de pruebas basada en equipos independientes o grupos de SQA. En el caso de que los desarrolladores sean los *testers* o equipos integrados de pruebas, este modelo de referencia resulta inapropiado, debido a la dimensión reducida de estos equipos la carga de trabajo sería excesiva. En el caso de las factorías de pruebas software no resulta apropiado ya que no contempla los procesos de gestión de servicios.

Este modelo, al igual que CMMi, tiene una gran desventaja a la hora de ser empelado por las organizaciones, puesto que no contiene una definición clara de los procesos, actividades y tareas a ejecutar, elementos de trabajo, productos de trabajo, etc. Por tanto, no existe una descripción formal del proceso; si no que simplemente proporciona información a ser considerada en la definición de los mismos a alto nivel. Este hecho frena su utilización en las organizaciones, ya que no proporciona la guía necesaria para la propia definición de los procesos de prueba (STEINER, M., BLASCHKE, M., PHILIPP, M., and SCHWEIGERT, T., 2010).

Por tanto, TMMi proviene de ser utilizado como modelo de referencia por las organizaciones para ser evaluadas con respecto a sus procesos de prueba y

determinar la madurez de los mismos, de forma que éstas puedan conocer de forma objetiva la eficacia de los mismos.

Además, la aplicación de TMMi está fuertemente ligada a CMMI, requiere que las organizaciones sigan CMMI en su representación por etapas como modelo de referencia para la mejora del resto de procesos no relacionados con las pruebas (LEE, C. 2009).

### **3.1.5 TMap - Test Management Approach**

TMap es una aproximación para la gestión de las pruebas que ha sido desarrollado por Sogeti. TMap considera pruebas dirigidas a resultados de productos software, tanto a alto como a bajo nivel. El método proporciona respuestas a preguntas sobre pruebas, como:

¿Qué?, ¿Cuándo?, ¿Cómo?, ¿Dónde?, y ¿Quién? (KOOMEN, T., VAN DER AALST, L., BROEKMAN, B., VROON, M., 2006).

El contenido específico que propone TMap considera cuatro elementos fundamentales:

- Gestión de pruebas dirigida al negocio (BDTM).
- Proceso de pruebas estructurado.
- Proporcionar las herramientas adecuadas.

La Figura 3-9 muestra una visión global del modelo:

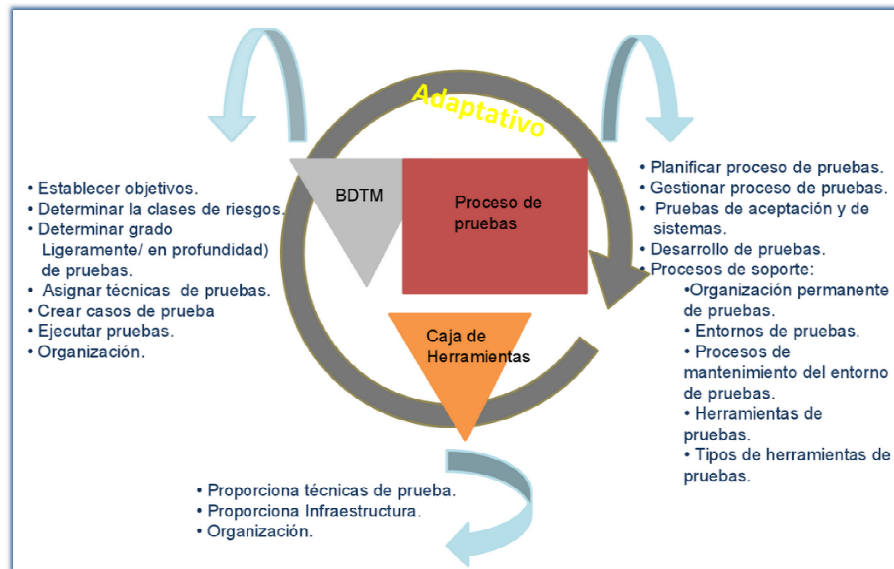


Figura 3-9 Modelo TMAP

A continuación se muestran de forma más detallada cada uno de los elementos fundamentales.

## 1. Gestión de pruebas dirigida al negocio (BDTM).

Para asegurar la utilidad de las pruebas se debe establecer, para el conjunto de pruebas, las características y las secciones de los objetos de prueba que representan un riesgo, para seleccionar aquellas que van a ser ejecutadas. Así, la selección dependerá de los riesgos que la organización piense que pueda incurrir, la cantidad de tiempo y dinero disponible, y el resultado que la organización desee alcanzar. Este hecho constituye la base de la propuesta Gestión de pruebas dirigida al negocio (BDTM – Business Driven Test Management) (KOOMEN, T., VAN DER AALST, L., BROEKMAN, B., VROON, M., 2006). Las propiedades específicas de BDTM son: el esfuerzo total está relacionado con los riesgos del sistema que va a ser probado, la estimación y planificación del proceso de pruebas está relacionada con la estrategia de pruebas definida, y hay que implicar al cliente en el proceso de pruebas.

## 2. Proceso de pruebas estructurado.

El proceso de pruebas estructurado se encuentra dividido en cuatro partes:

**Plan de pruebas maestro:** El director de pruebas junto con el cliente y otros *stakeholders* que se encuentren involucrados deberán establecer qué se debe probar en el sistema y con qué grado de profundidad; con el objetivo de detectar el mayor número de defectos importantes lo antes posible y con el menor costo (SOGETI, 2008b).

Pruebas de aceptación y de sistema: Las pruebas de aceptación y sistema son consideradas como procesos autónomos que deben ser organizados; por lo que tienen su propio plan de pruebas, presupuesto y, frecuentemente, su entorno de pruebas propio. Se ejecutan en paralelo al proceso de desarrollo que debe comenzar mientras se crean las especificaciones funcionales. TMap define un modelo de ciclo de vida genérico para las pruebas de aceptación y sistema, que consta de siete fases:

Planificación, Control, Organización y mantenimiento de la infraestructura, Preparación, Especificación, Ejecución y Finalización. La figura siguiente ilustra este modelo (SOGETI, 2008b).

**Pruebas de desarrollo:** Las pruebas de desarrollo utilizan conocimiento de la implementación técnica del sistema, comenzando por los primeros elementos que se han desarrollado o los más pequeños como rutinas, unidades, programas, módulos, objetos, etc.; sin embargo no constituyen una parte integral del trabajo del desarrollador. La elaboración detallada de las mismas puede variar por proyecto u organización y depende, entre otras cosas, del método de desarrollo utilizado y la disponibilidad de ciertas medidas de calidad, como por ejemplo la calidad acordada (SOGETI, 2008b).

**Procesos de soporte:** Para realizar las pruebas se hace necesario un entorno de pruebas que soporte a las mismas, y que en algunos casos cumpla con un conjunto de requisitos genéricos que garanticen la ejecución fiable de las pruebas. Debe ser representativo, manejable y flexible, y también debe garantizar la continuidad de las pruebas. Además, para prevenir problemas en el entorno de

pruebas es necesaria la implantación de procesos para gestionar la configuración y el mantenimiento de entornos. Estos procesos pueden ser: Gestión de configuración, Gestión del cambio, Gestión de incidencias, Gestión de problemas, y Gestión de datos (SOGETI, 2008b).

### **3. Proporcionar las herramientas adecuadas.**

TMap soporta la ejecución del proceso de pruebas estructurado mediante una caja de herramientas completa, centrada en los temas siguientes (SOGETI, 2008c):

**Técnicas:** Cómo se prueba el proceso. Técnicas de análisis de riesgo, de estimación, de evaluación, etc.

**Infraestructura:** Dónde y con qué es probado. Ésta se compone del entorno de pruebas, las herramientas de pruebas, y el lugar de trabajo.

**Organización:** Quién hace las pruebas. Al comienzo de un proyecto de pruebas se deben definir y asignar los diferentes roles implicados, las tareas a realizar, y las autorizaciones y responsabilidades que se consideren necesarias. Esto podrá ser realizado para el proceso de pruebas total o para un nivel específico.

TMap, al igual que TPI, podría ser utilizado en organizaciones que utilizan una estructura organizativa de pruebas basada en equipos independientes o grupos de SQA. En el caso de que los desarrolladores sean los *testers* o equipos integrados de pruebas, dónde los equipos no son muy numerosos, la carga de trabajo sería excesiva. En relación a las factorías de pruebas software, TMap define un modelo que incluye procesos dirigidos a la gestión del servicio. Sin embargo, requiere de tener implantada la metodología TMap, los procesos no están definidos en detalle y no se describe la interacción entre los mismos.

La principal limitación que aparece a la hora de implementar TMap es la extensión que tiene. Es una metodología que recoge una gran cantidad de prácticas provenientes de la experiencia de consultores en el área de pruebas de software, lo que proporciona una información muy detallada. Sin embargo, como

consecuencia de la extensión de esta aproximación, su adaptación a muchas de las estructuras de pruebas es muy complicada, incluso en los escenarios vistos previamente, equipos independientes o grupos de SQA, salvo que se trate de grandes equipos con múltiples recursos.

Además, la cantidad de procesos que define, roles y actividades a llevar a cabo es más adecuada a grandes organizaciones o departamentos que se dedican únicamente a la realización de actividades de verificación y validación de productos software. Su adaptación a entornos de pequeñas y medianas empresas es complejo y requiere de un trabajo de consultoría que permita adaptar los procesos, roles y actividades a llevar a cabo a las particularidades de este tipo de organizaciones.

Como se ha visto en el apartado anterior, para poder determinar acciones de mejoras en el proceso de pruebas, TMap deber ser complementado por TPI, que es el mecanismo definido por Sogeti para este propósito. TMap en sí mismo no proporciona medios para la evaluación y mejora del proceso de pruebas.

#### **3.1.5.1 Roles TMap**

TMap otorga una especial importancia al concepto de rol, si bien hasta ahora se ha venido considerando como un concepto de apoyo para dar a un perfil profesional una mayor especialización dentro de una organización evitando su equiparación. En TMap el concepto de rol se centra en las actividades de prueba que son realizadas dentro de un proyecto específico de la organización. Por ello, forman un conjunto de competencias mucho más especializadas que los recogidos en las posiciones o perfiles profesionales, pero así mismo, al ser en el ámbito de una misma organización se encuentran fuertemente relacionados. Llegando a existir, roles que se corresponden con las posiciones mencionadas en el apartado anterior, como son (DRIEL, 2010):

- Tester.
- Programador de herramientas de pruebas.
- Experto en métodos de prueba.

- Coordinador de pruebas.
- Experto en herramientas de pruebas.
- Analista de pruebas.
- Gerentes de pruebas.
- Consultor de herramientas de pruebas.

En Tmap Next por ejemplo, propone una serie de roles que no concuerdan en su totalidad con ninguna posición a modo de guía, tal y como se puede apreciar a continuación (DRIEL, 2010):

- **Integrador de aplicaciones:** El integrador de aplicaciones es el responsable de la integración de las diferentes partes del sistema (programas, objetos, módulos y componentes) en un sistema operativo correctamente, y es capaz de situarse como un inspector de calidad a fin de supervisar los acuerdos alcanzados. El integrador de aplicaciones apoya a la unidad de pruebas y lleva a cabo la unidad de integración de pruebas. Además prepara el sistema para la siguiente fase del proyecto.

Competencias específicas:

- ✓ La integración de los distintos programas, objetos, módulos y componentes en el sistema.
- ✓ La obtención de la aprobación y del mantenimiento de la unidad de integración del plan de pruebas.
- ✓ La creación de los criterios de salida que una parte integrada del sistema debería tener para estar preparada para afrontar la siguiente fase.
- ✓ El soporte a los programadores en la ejecución de las pruebas de unidad.
- ✓ La dirección de la ejecución de la gestión de la configuración y la gestión de la versión.
- ✓ La dirección de la ejecución de la gestión de defectos internos.
- ✓ El contacto con cliente sobre los tipos de pruebas posteriores.
- ✓ La información del progreso de los procesos de integración y de la calidad de los objetos de prueba.
- ✓ El asesoramiento sobre la unidad de pruebas de integración.
- ✓ La evaluación de los procesos de integración.
- ✓ La preparación del sistema para la siguiente fase.

- **Administrador de defectos:** El administrador de defectos es responsable de configurar y optimizar el uso de la administración de defectos y de los entornos de los procesos.

Competencias específicas:

- ✓ La configuración de la administración de los defectos y de los procesos asociados.
- ✓ El asesoramiento sobre la selección de herramientas para la administración de defectos.
- ✓ El seguimiento, la evaluación y la mejorar de los procesos y de los métodos en administración de defectos.
- ✓ La gestión de los registros de defectos mediante el análisis y los foros de toma de decisión.
- ✓ La administración del acceso a las herramientas de la administración de defectos.

- **Experto en el conocimiento del dominio:** El experto en el conocimiento del dominio ofrece un conocimiento sobre la materia y una visión de cómo el dominio está especificado en los requerimientos, diseños funcionales, procesos de negocio y en los manuales de usuario.

Competencias específicas:

- ✓ El soporte y el asesoramiento sobre los procesos funcionales y de negocio.

- **Intermediario:** El intermediario mantiene el contacto a nivel operacional con el equipo de pruebas, con los desarrolladores y con los expertos en la materia. Por otro lado, explican los defectos y proporcionan datos complementarios si son necesarios. También suministran soluciones a los defectos de los equipos de pruebas. Este rol toma especial importancia en el caso de existir equipos de pruebas distribuidos o si existe algún tipo de relación formal con otra parte interesada, como en el caso del outsourcing.

Competencias específicas:

- ✓ La exposición de los defectos a los consultores de calidad y a los participantes en el foro de análisis.
- ✓ La filtración defectos duplicados.
- ✓ La suministración datos complementarios a los defectos.



- ✓ La dirección de la evaluación de los defectos cuando estos se pueden producir bajo otros entornos.
- ✓ La comprobación de las soluciones para el ajuste de la política y la integridad del proyecto.
- ✓ La exposición de la solución a los defectos.
- ✓ El seguimiento del estado de los defectos.
- ✓ El informe del progreso de los defectos.
- ✓ La asesoría a los órganos de consultaría sobre los defectos.
- **Experto de sistemas:** El experto en sistemas tiene la responsabilidad de explicar a los probadores la funcionalidad y la arquitectura técnica en general del sistema bajo el que prueban. Debe dar soporte a las pruebas de sistema, a las pruebas de aceptación funcional, a las pruebas de aceptación de usuario y a las pruebas de integración de sistema.

Competencias específicas:

- ✓ La exposición de la funcionalidad del sistema, también en relación con el proceso de negocio.
- ✓ La exposición de la arquitectura técnica del sistema.
- **Coordinador de infraestructuras de pruebas:** El coordinador de infraestructuras de pruebas es responsable dentro del equipo de pruebas del contacto con las partes encargadas de configurar y mantener un alto nivel de disponibilidad de la infraestructura de las pruebas (entorno de pruebas, herramientas de pruebas y el lugar de trabajo del probador). Este rol se desarrolla preferiblemente por una única persona, preferiblemente a tiempo parcial. Si no es posible encontrar la persona idónea para este rol las tareas pueden ser distribuidas.

Competencias específicas:

- ✓ El aseguramiento de la disponibilidad oportuna de los entornos de prueba
- ✓ El aseguramiento de la disponibilidad oportuna de las herramientas de pruebas con la planificación y el control requeridos tan pronto como lo requieran las herramientas de ejecución de pruebas.
- ✓ El aseguramiento de la disponibilidad oportuna de los puestos de trabajo.
- ✓ La coordinación con otras partes de la infraestructura de las pruebas.

- ✓ El asesoramiento sobre la resolución de defectos o incidentes en el área de la infraestructura de las pruebas.
- ✓ El asesoramiento sobre el impacto de los cambios propuestos en la infraestructura de las pruebas.
- ✓ El asesoramiento sobre las especificaciones de la infraestructura de las pruebas.
- ✓ El asesoramiento sobre los costos del uso de una infraestructura de pruebas.
- **Administrador del proyecto de pruebas:** El administrador del proyecto de pruebas es responsable de la administración de la información y de la disponibilidad del proyecto de pruebas, incluyendo la información del progreso y la documentación del proyecto.  
Competencias específicas:
  - ✓ La recogida, la comprobación y el registro de la información del progreso.
  - ✓ La suministración de la información del progreso.
  - ✓ La recogida y la administración la documentación del proyecto.
- **Líder del equipo de pruebas e Inspecciones:** El líder del equipo de pruebas e inspecciones es el responsable del día a día del trabajo del equipo de calidad. Este rol se introduce si el equipo de pruebas e inspecciones se encuentra distribuido. Por lo cual, cada equipo es dirigido por un líder, aunque si el equipo de pruebas e inspecciones es muy pequeño este rol se combina con el rol de un tester.  
Competencias específicas:
  - ✓ La creación detallada de la planificación del equipo de pruebas.
  - ✓ El seguimiento e el informe del progreso del proceso de pruebas.
  - ✓ El informe de la calidad del objeto de pruebas.
  - ✓ La gestión de la resolución de problemas de infraestructura.
  - ✓ El informe de cuellos de botella y de sus soluciones.
  - ✓ La gestión diaria del equipo de pruebas e inspecciones.
  - ✓ La formación de probadores en el equipo de pruebas e inspecciones.

- **Administrador** de los programas de pruebas: El administrador de los programas de pruebas es el responsable del seguimiento y del cumplimiento de las regulaciones existentes en el área de los programas de prueba.

Competencias específicas:

- ✓ La configuración de los programas, los procedimientos y las regulaciones de las pruebas.
- ✓ La gestión del programa de pruebas.
- ✓ La configuración de la gestión del programa, de la configuración de los básicos, de los objetivos y de la infraestructura de las pruebas.
- ✓ Comprobación del cumplimiento con la regulación de programa de pruebas.

#### **3.1.5.2 Gestión de Defectos**

Una buena administración debe ser capaz de controlar el ciclo de vida de un defecto y ofrecer distintas visiones generales. Esas visiones se utilizan, entre otras cosas, para hacer declaraciones de calidad bien fundadas (TMAP, 2010).

El equipo de V&V debe tener en cuenta que se encuentran juzgando el trabajo de otras personas y que el producto final es el resultado de la colaboración entre todas las partes implicadas en el mismo. Por ello, el equipo de V&V debe adoptar una postura lo más neutral posible para encontrar defectos.

En el proceso de pruebas, los probadores informan de la presencia de defectos y comprueban sus soluciones. Pero, es el gerente de pruebas quien se encarga de interactuar con el grupo implicado en el defecto, preservando así la independencia entre los dos grupos (TMAP, 2010).

#### **3.1.5.3 Métricas**

La definición, el mantenimiento y el uso de métricas son importantes para el proceso de pruebas, ya que permite al gerente de pruebas comunicar el estado de éstas a través de hechos (MARTIN, 2002). TMap ofrece instrucciones de cómo establecer un conjunto práctico de iniciación a las métricas de las pruebas basado en el método de las seis etapas de métrica-pregunta-objetivo, las cuales son:

- Definición de objetivos.
- Realizar preguntas sobre los defectos por objetivo.
- Realizar preguntas sobre las posibles métricas a aplicar.
- Recoger información del proceso de pruebas y analizar.

- Presentar y distribuir los datos de medida.
- Relacionar los datos de medida con las preguntas y objetivos.

Además, TMap proporciona listas de métricas útiles para hacer valoraciones sobre la calidad del objeto y del proceso de prueba:

- Números de defectos encontrados. Ratio entre el número de defectos encontrados y el tamaño del sistema por unidad del tiempo de prueba.
- Instrucciones ejecutadas. Proporción del número de instrucciones del programa probadas por el número total de instrucciones.
- Pruebas de número. Proporción entre las pruebas y el tamaño del sistema. Indica cuantas pruebas son necesarias en una parte.
- Numero de caminos probados. Proporción entre el número de rutas probadas y su total.
- Numero de defectos durante la producción. Esta métrica indica el número de defectos no encontrados durante el proceso de prueba.
- Efectividad de la detección de defectos. Número total de defectos encontrados durante las pruebas, dividido entre el total de defectos estimados.
- Costo de las pruebas. Proporción entre los costos de las pruebas y los costos del desarrollo.
  - ✓ Costo por defecto localizado. Costo total de las pruebas entre el número total de defectos encontrados.
  - ✓ Presupuesto utilizado. Proporción del presupuesto y el costo actual de las pruebas.
  - ✓ Eficiencia de las pruebas. Número de pruebas requeridas entre el número de defectos localizados.
  - ✓ Grado de automatización de las pruebas. Proporción entre el número de pruebas llevadas a cabo de forma manual y el número de pruebas automáticas.
  - ✓ Numero de defectos encontrados (relativos). La proporción entre el número de defectos encontrados y el tamaño del sistema por unidad del tiempo de prueba.

- ✓ Defectos como resultados de las modificaciones no probadas. Defectos a causa de las modificaciones que no están probadas, como una parte del total del número de defectos derivados como resultados de los cambios.
- ✓ Defectos después de las modificaciones probadas. Defectos a causa de las modificaciones que están probadas, como una parte del total del número de defectos derivados como resultados de los cambios.
- ✓ Ahorros de las pruebas. Indicador de los ahorros provocados por las pruebas.

### **3.1.6 MoProSoft**

El modelo de procesos MoProSoft tiene tres categorías de procesos: Alta Dirección, Gerencia y Operación que reflejan la estructura de una organización. La categoría de Alta Dirección contiene el proceso de Gestión de Negocio. La categoría de Gerencia está integrada por los procesos de Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos.

Éste último está constituido por los subprocesos de Recursos Humanos y Ambiente de Trabajo, Bienes, Servicios e Infraestructura y Conocimiento de la Organización.

La categoría de Operación está integrada por los procesos de Administración de Proyectos Específicos y de Desarrollo y Mantenimiento de Software.

En cada proceso están definidos los roles responsables por la ejecución de las prácticas. Los roles se asignan al personal de la organización de acuerdo a sus habilidades y capacitación para desempeñarlos.

En MoProSoft se clasifican los roles en Grupo Directivo, Responsable de Proceso y otros roles involucrados. Además se considera al Cliente y al Usuario como roles externos a la organización.

#### **3.1.6.1 Categoría de procesos**

Categoría de Alta Dirección (DIR): Categoría de procesos que aborda las prácticas de Alta Dirección relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos.

**Categoría de Gerencia (GER):** Categoría de procesos que aborda las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Proporciona los elementos para el funcionamiento de los procesos de la Categoría de Operación, recibe y evalúa la información generada por éstos y comunica los resultados a la Categoría de Alta Dirección.

**Categoría de Operación (OPE):** Categoría de procesos que aborda las prácticas de los proyectos de desarrollo y mantenimiento de software. Ésta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y entrega a ésta la información y productos generados. A continuación en la Figura 3-10 se muestra el Diagrama de categorías de procesos Moprosoft.

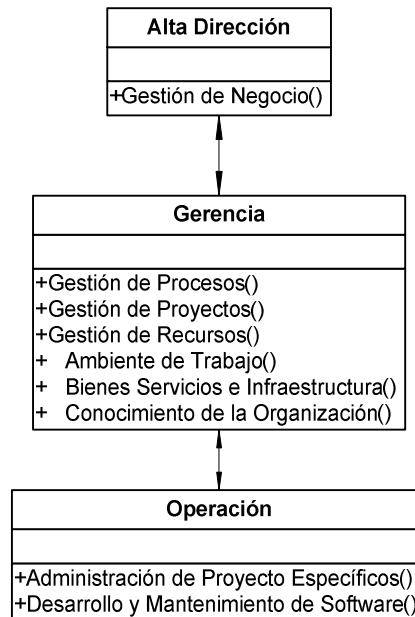


Figura 3-10 Diagrama de categorías de procesos Moprosoft

### 3.1.6.2 Procesos

**DIR.1 Gestión de Negocio:** El propósito es establecer la razón de ser de la organización, sus objetivos y las condiciones para lograrlos, para lo cual es necesario considerar las necesidades de los clientes, así como evaluar los resultados para poder proponer cambios que permitan la mejora continua. Adicionalmente habilita a la organización para responder a un ambiente de cambio y a sus miembros para trabajar en función de los objetivos establecidos.

GES.1 Gestión de Procesos: El propósito es establecer los procesos de la organización, en función de los procesos requeridos identificados en el plan estratégico, así como definir, planificar, e implantar las actividades de mejora de los mismos.

GES.2 Gestión de Proyectos: El propósito es asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización.

GES.3 Gestión de Recursos: El propósito es conseguir y dotar a la organización de los recursos humanos, infraestructura, ambiente de trabajo y proveedores, así como crear y mantener la base de conocimiento de la organización. La finalidad es apoyar el cumplimiento de los objetivos del plan estratégico de la organización.

GES.3.1 Recursos Humanos y Ambiente de Trabajo: El propósito es proporcionar los recursos humanos adecuados para cumplir las responsabilidades asignadas a los roles dentro de la organización, así como la evaluación del ambiente de trabajo.

GES.3.2 Bienes, Servicios e Infraestructura: El propósito es proporcionar proveedores de bienes, servicios e infraestructura que satisfagan los requisitos de adquisición de los procesos y proyectos.

GES.3.3 Conocimiento de la Organización: El propósito es mantener disponible y administrar la base de conocimiento que contiene la información y los productos generados por la organización.

OPE.1 Administración de Proyectos Específicos: El propósito es establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados.

OPE.2 Desarrollo y Mantenimiento de Software: El propósito es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software, nuevos o modificados, cumpliendo con los requerimientos especificados. A continuación en la Figura 3-11 se muestra el Diagrama de relación entre procesos Moprosoft.

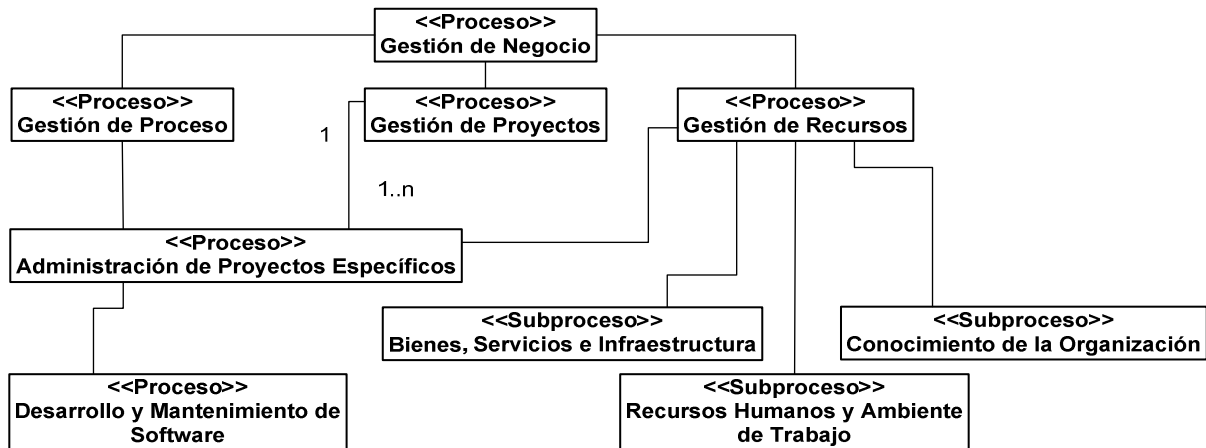


Figura 3-11 Diagrama de relación entre procesos Moprosoft

### 3.2 Comparación de Modelos de Calidad

Luego de analizar los modelos existentes como CMMI, TMMI, TMap, Moprosoft y Competissoft; se puede afirmar que todas consideran el proceso de Verificación y Validación de Software en el desarrollo de sus propuestas de mejora de procesos y por tanto concluir que se reconoce la importancia de que las empresas incluyan el Verificación y Validación de Software en sus procesos de desarrollo de SW.

CMMi con respecto al proceso de Verificación y Validación de Software propone 02 Áreas de Procesos: Verificación (VER) y Validación (VAL), describe las mejores prácticas a considerar pero sin ofrecer una guía para su implementación que sirva como referencia para las empresas que desean adoptarlo. El enfoque anterior deja a consideración y criterio de la empresa la implementación de las actividades de Verificación y Validación de Software en sus procesos, esta situación genera que las empresas contraten consultores especialistas para su implementación, que por su alto costo puede resultar prohibitivo para las PYMES. En Perú esto puede llegar a costar entre \$16,000.00 y \$30,000.00 dólares americanos. Otro factor a considerar es que la implementación puede tardar entre 12 y 18 meses.



El caso de TMMI el problema se reduce a que no existe en Perú una empresa o consultor especialista que facilite la adopción del modelo de mejora (<http://www.tmmi.org/?q=asesor>). También se debe considerar que las 07 (siete) empresas proveedoras acreditadas se ubican geográficamente en Europa (<http://www.tmmi.org/?q=node/48>), lo que reduce la posibilidad que una PYME adopte el modelo de mejora.

La problemática con respecto a TMAP es bastante parecida a la de TMMI, donde las empresas y especialistas de entrenamiento se ubican geográficamente en ciertos países de Europa (<http://www.tmap.net/training>).

El caso de las normas ISO se asemeja mucho al de CMMI, siendo el factor del costo un tema recurrente que dificulta la implementación de la norma por parte de las PYMES (Saiedian, 1997). El caso particular de los estándares de la IEEE es que la propuesta generalmente se orienta a grandes empresas, dificultando la adopción por parte de las PYMES.

En la Tabla 3.1 se muestra un cuadro comparativo sobre los modelos de calidad más representativos del mercado, pero enfocados en los procesos de Verificación y Validación de Software, de estos cuatro modelos se tomará en cuenta las mejores prácticas que serán aplicadas a una Pyme.

Característica / Estándar	CMMI	COMPETISOFT	MOPROSOFT	TMMI	TMap
Mejora de	Desarrollo de SW		Pruebas de Software		

procesos para					
Orientado a	Empresas de desarrollo de SW	Pequeñas y medianas empresas de Desarrollo de SW	Empresas de desarrollo de SW	Empresas de desarrollo de SW	Empresas de desarrollo de SW
Procesos de Validación	Área de Proceso específica para Validación (Nivel 3 - validación)	Define actividades de Validación para todos sus procesos (Gestión de Negocio, Gestión de Procesos, Gestión de Recursos, Gestión de Recursos Humanos, Administración de Proyectos, Desarrollo de Software, Mantenimiento de Software)	Define actividades de Validación en los procesos propuestos	Define "Prácticas Específicas" de Validación en "Metas" de las "Áreas de Proceso".	Define actividades de Validación en los procesos propuestos
Procesos de Verificación	Área de Proceso específica para Verificación (Nivel 3 - verificación)	Define actividades de Verificación para todos sus procesos (Gestión de Negocio, Gestión de Procesos, Gestión de Recursos, Gestión de Recursos Humanos, Administración de Proyectos, Desarrollo de Software, Mantenimiento de Software)	Define actividades de Verificación en los procesos propuestos	Define "Prácticas Específicas" de Verificación en "Metas" de las "Áreas de Proceso".	Define actividades de Verificación en los procesos propuestos
Implementación	Se implementan los Niveles a través de las Áreas de Proceso especificadas para cada Nivel de Madurez	Se implementa a través de la definición de Procesos, Actividades, Roles y Herramientas	Se implementa a través de la definición de Procesos, Actividades y Roles. Se implementan los Niveles a través de las Áreas de Proceso especificadas para cada Nivel de Madurez	Se implementan los Niveles a través de las Áreas de Proceso especificadas para cada Nivel de Madurez	Se implementa a través de la definición de Procesos, Actividades y Roles

Tabla 3-1 Cuadro comparativo de Procesos de Calidad enfocados al proceso de V&V (Elaboración Propia)

### **3.3 Artículos relacionados**

#### **3.3.1 “Integration, Verification, Validation, Test And Evaluation (IVVT&E) Framework for System of Systems (SoS)”**

Cuando se dice que existe SoS es porque en esta ingeniería se sigue en su mayoría los siguientes atributos: Gestión operacional e independiente, Distribución Geográfica, Comportamiento emergente y Desarrollo evolutivo. Una gran preocupación y desafío mencionada es la escalabilidad de las metodologías de testeo para este tipo de ingeniería, sobre todo cuando un gran número de sistemas envueltos en el marco de trabajo IVVT&E son muy costosas o implican consumo de tiempo en un periodo limitado (LUNA, SERGIO; LOPES, AMIT; YAN SEE TAO, HOONG; ZAPATA, FRANCISCO; PINEDA, RICARDO, 2013).

Por tal motivo, la lectura nos habla de metodologías que son una extensión de los principios de la ingeniería de sistemas y que estos servirán de enlace para identificar los desafíos mencionados, por ejemplo: Department of Defense Architecture Framework (DoDAF) sirve para identificar entidades relevantes del SoS y sus interfaces, Design Structure Matrix (DSM) sirve para optimizar el flujo de la información entre los sistemas y estrategias de testeo combinatorios que sirven para optimizar los tests y evaluación del SoS (LUNA, SERGIO; LOPES, AMIT; YAN SEE TAO, HOONG; ZAPATA, FRANCISCO; PINEDA, RICARDO, 2013a).

Al hacer un breve recuento de lo mencionado anteriormente se puede mencionar el Department of Defense Architecture Framework (DoDAF), el cual es un marco de arquitectura basado en los requerimientos más entendidos, de tal forma que asegura que todos los requerimientos están siendo considerados de tal modo que se encuentren alienados.

En el caso de las interfaces de los SoS, la selección de los estándares que son ampliamente usados es importante porque de esta forma se habilita los sistemas que van a ser reemplazados con mínima re arquitectura. Por otro lado, las capas de las interfaces sirven para complementar los estándares. En cuanto al Design Structure Matrix (DSM) aplicado a un UAS obtiene una representación gráfica de las actividades y procesos del framework IVVT&E a analizar del modelo formal del sistema en una concisa y compacta manera que es fácil de entender (LUNA, SERGIO; LOPES, AMIT; YAN SEE TAO, HOONG; ZAPATA, FRANCISCO; PINEDA, RICARDO, 2013b).

Finalmente, la lectura propone el Framework IVVT&E y utiliza la plataforma de comunicaciones de un Unmanned Aerial System (UAS) como ejemplo.

### **3.3.2 Metodología para implantar el Modelo Integrado de Capacidad de Madurez en grupos pequeños y emergentes.**

El artículo presenta como es posible hacer crecer competitivamente a las pymes de software aplicando un modelo CMMI (CapabilityMaturity Model Integration) a partir de la metodología SPI (Software Process Improvement). Esta propuesta que es aplicada a estas empresas emergentes también está basada sobre otro modelo llamado IDEAL (Initiating, Diagnosing, Establishing, Acting and Learning). Al inicio, muchos influyentes en el tema creían que este modelo debía ser aplicado únicamente a empresas grandes, donde el conocimiento y madurez es mayor, además de que muchas pymes se negaban a este modelo por temas de dinero, infraestructura y gastos en capacitación a su personal por conocimientos que faltaban, pero eso ha sido refutado con las buenas prácticas que han sido aplicadas en muchas empresas pequeñas que si aceptaron el modelo durante el tiempo. Otros personajes en el artículo indican que la mejor

manera de aplicar CMMI es uniéndolo con SCRUM. Las metodologías ágiles como: MoProSoft, ISO, QFD y TSP muestran que al implementar mejora de procesos en entornos pequeños, estas se benefician en gran medida y de manera incremental por pequeños cambios en sus procesos (ARBOLEDAA, HUGO; PAZBY RUBBY CASALLAS, ANDRÉS, 2013).

Centrándonos de regreso al modelo CMMI, este se basa en tres principios: pequeños pasos para el mejoramiento alineados con los proyectos de desarrollo; procesos livianos apoyados en herramientas open source y visibilidad a corto plazo de los beneficios. La metodología incluye: la definición de un modelo de mejoramiento organizacional; una matriz de grados de logro de las prácticas específicas de CMMI, y un mapa de ruta donde se están definidas los procesos y herramientas de apoyo para facilitar la marcha continua. El hecho de aplicar el modelo CMMI a una pyme, no significa que necesariamente va a crecer, ya que esta junto con sus integrantes deben de cumplir algunas condiciones que deben ser realizadas con esfuerzo y empeño (ARBOLEDAA, HUGO; PAZBY RUBBY CASALLAS, ANDRÉS, 2013).

Para lograr comprender un poco más de este modelo se explicará sobre la matriz de logro y el mapa de ruta.

Matriz de Logro: Una de las sugerencias del CMMI es implementar prácticas a ser cumplidas por los grupos de trabajo para poder definir un conjunto de logros. Estas prácticas contienen un conjunto de actividades. Cuanto mayor sea el grado del logro a cumplir, más completo estará conjunto de actividades y mayor será la satisfacción (ARBOLEDAA, HUGO; PAZBY RUBBY CASALLAS, ANDRÉS, 2013).

Mapa de Ruta: Son las estrategias que facilitan la implantación de las prácticas específicas propuestas.

### **3.3.3 Developing an analytical model for planning systems verification, validation and testing processes**

El VVT o sistema de verificación, validación y pruebas tiene como objetivo principal garantizar que los sistemas están diseñados y que cumplen de la mejor manera con los requerimientos establecidos por el cliente, logrando su satisfacción.

Se realizaron varios estudios con el fin de mejorar el proceso de VVT. El Dr. Engel desarrolló métodos cuantitativos para modelar y estimar el costo y riesgo de ciertas estrategias, que forman parte del proceso, así como optimizar el mismo (LUNA, SERGIO; LOPES, AMIT; YAN SEE TAO, HOONG; ZAPATA, FRANCISCO; PINEDA, RICARDO, 2013b).

El ingeniero de Sistemas define subjetivamente el riesgo involucrado en cada proyecto que lidera, mediante la elección de ciertos métodos de verificación. Los riesgos en algunos casos son percibidos como recursos propios del proyecto (LUNA, SERGIO; LOPES, AMIT; YAN SEE TAO, HOONG; ZAPATA, FRANCISCO; PINEDA, RICARDO, 2013b).

El desarrollo de software es muy diferente al desarrollo de otros productos que poseen componentes de hardware. Por lo tanto, el proceso de VVT y sus actividades son distintas.

R. Dolores, en uno de sus estudios, describe diferentes técnicas o métodos para realizar las actividades de V&V, seleccionadas de tres grandes grupos: estático, dinámico y formal.

La técnica de análisis estático analiza la forma y estructura. La técnica de análisis dinámico se define como la simulación realizada por el producto con el fin de detectar errores mediante el análisis de las respuestas que brinda. Y la técnica de análisis formal, incluye el uso de técnicas matemáticas, hace uso de algoritmos (LUNA, SERGIO; LOPES, AMIT; YAN SEE TAO, HOONG; ZAPATA, FRANCISCO; PINEDA, RICARDO, 2013a).

El proceso de VVT debe realizar durante todo el ciclo de vida del producto. Algunos encargados del proyecto consideran la aplicación del proceso de VVT como un camino que conduce al éxito, mientras que otros lo ven como un obstáculo innecesario. Aunque la aplicación de este proceso no necesariamente garantiza el éxito, su beneficio se ve reflejado en el resultado de los métodos y herramientas utilizadas, y los recursos asignados.

Todo proceso de VVT debe tener en cuenta lo siguiente:

- Clientes/Stakeholders
- Requerimientos no funcionales y funcionales
- Estrategias propias del VVT
- Actividades propias del VVT
- Y métodos, para verificar y validar las estrategias y actividades del proceso.

#### **3.3.4 25 años de Investigacion Testing en Brazil.**

En los últimos 25 años, el Simposio Brasileño de Ingeniería de Software (SBES) ha sido participe de los eventos más importantes en la ingeniería de software (SE). En estos últimos años, el SBES ha reunido una gran cantidad de estudios en las pruebas de software. Este fue dirigido a



proporcionar un entendimiento profundo de lo que ya se había publicado, en este caso, se han sintetizado sus 25 años de historia de la investigación sobre las pruebas de software. El uso de la información obtenida de este resumen destacamos que los temas de pruebas de software han sido los más ampliamente estudiado en la literatura (VINICIUS HUMBERTO SERAPILHA DURELLIA; RODRIGO FRAXINO ARAUJOA; MARCO AURELIO GRACIOTTO SILVA; RAFAEL ALVES PAES DE OLIVEIRAA; JOSE CARLOS MALDONADOA; MARCIO EDUARDO DELAMAROA, 2012).

El SBES ofrece una recopilación de las herramientas de pruebas de software como:

- PROTEUM, que se ha sido muy utilizado, el más común y conocido por los investigadores de todo el mundo.
- JABUTI, una herramienta que ha sido para la comprensión y las pruebas utilizando para la investigación de pruebas estructurales.

Concepto del estudio sistemático de mapeo, es una cartografía y un método de síntesis que implica una búsqueda para determinar la naturaleza, el alcance y la cantidad de trabajos de investigación publicados en una determinada área de interés (VINICIUS HUMBERTO SERAPILHA DURELLIA; RODRIGO FRAXINO ARAUJOA; MARCO AURELIO GRACIOTTO SILVA; RAFAEL ALVES PAES DE OLIVEIRAA; JOSE CARLOS MALDONADOA; MARCIO EDUARDO DELAMAROA, 2012).

Estudios de mapeo de un agregado y categoriza los estudios primarios dando una vista sintetizada de una cierta área de investigación. Estos contribuyen a la replicabilidad y la transparencia (VINICIUS HUMBERTO SERAPILHA DURELLIA; RODRIGO FRAXINO ARAUJOA; MARCO

AURELIO GRACIOTTO SILVA; RAFAEL ALVES PAES DE OLIVEIRAA; JOSE CARLOS MALDONADOA; MARCIO EDUARDO DELAMAROA, 2012).

El documento es una extendida versión del estudio de mapeo. Tiene como objetivo: Identificar las áreas de investigación de pruebas de software.

Por otro lado, el documento de los 25 años de investigación testing en Brazil explica acerca de un procedimiento que se debe seguir para un correcto testing.

### **Seccion2.** Describir el estudio del mapeo.

Se encuentran en 4 pasos:

- Definir las preguntas de investigación.
- Pantallas de papel.
- Extracción de data y mapeo.
- Evalúa las preguntas correspondientes antes de ser impactadas por la investigadora brasilera de prueba de software.

### **Seccion3.** Clasificación de acuerdo al tipo de estudio.

De acuerdo al autor, existen 5 categorías con respecto al tipo de estudio:

- Validación de investigación.
- Evaluación de la investigación.
- Proponer soluciones.
- Opiniones en papel.
- Experiencia en papel.

Clasificar el análisis de estudio en 4 categorías:

- Proponer solución (es aquella solución técnica, las más usada, efectiva y relevante)

- Evaluación de investigación (está enfocada en la evaluación de un problema)
- Validación de la investigación (estudio de la propuesta de solución investigada)
- Opinión (aquella que contiene los puntos de vista del autor.)

**Seccion4.** Clasificación de la prueba de software.

De acuerdo a la contribución del estudio primario se divide en tres categorías:

- Tecnología.
- Origen de la información.
- Fase de pruebas.

**Sección5.** Presentar un detalle de como los grupos de investigación de sus principales roles en la historia de SBES.

**Seccion6.** Estimar la revelación de SBES como 25 años de impacto sobre las pruebas de esfuerzo del software.

**Seccion7.** La información recolectada es comparada con la investigación del SBES.

**Seccion8.** A continuación esos temas de investigación de pruebas estén direccionadas hacia el futuro.

### **3.3.5 A systematic mapping study on the combination of static and dynamic quality assurance techniques**

En el siguiente documento, se encontrara información acerca de las técnicas que se utilizan para obtener productos de alta calidad. También, se

realizará un estudio de mapeo sistemático sobre la combinación de las diferentes técnicas de gestión de calidad. Mediante este estudio, se demuestra la importancia de tener un control constante durante el desarrollo de un producto, desde que este empieza (FRANK ELBERZHAGER JÜRGEN MÜNCH; VI TRAN NGOC NHA, 2012).

Actualmente, existen técnicas y métodos de control de calidad estáticas y dinámicas. Las técnicas estáticas se encargan de examinar todos los artefactos o documentos, con los que cuenta la empresa, para realizar la gestión de calidad. Esta técnica no necesita el programa. Por otro lado, las técnicas dinámicas necesitan el programa completo o parte del programa que se está analizando para realizar la gestión de calidad. Ambas técnicas, se deben realizar en tiempos determinados del desarrollo. Primero, durante las fases tempranas del desarrollo, se debe realizar métodos estáticos. Para culminar, durante las fases siguientes del desarrollo, se debe realizar métodos dinámicos (FRANK ELBERZHAGER JÜRGEN MÜNCH; VI TRAN NGOC NHA, 2012).

La realización de la gestión de calidad de los productos tiene una gran importancia a lo largo de la etapa de desarrollo como la de planeamiento y termino del producto. Sin embargo, mediante estudios previamente realizados, se ha podido observar y verificar que existen casos en donde no se realizan estas técnicas de gestión de calidad. Además, en los casos donde se realizan no se les da la prioridad necesaria, causando que esta evaluación se realice al culminar el producto. Es por ello, que se propone la combinación de ambas técnicas (FRANK ELBERZHAGER JÜRGEN MÜNCH; VI TRAN NGOC NHA, 2012).

A continuación, para analizar y decidir que es óptimo la combinación de dichas técnicas, se comenzará a realizar un estudio de recopilación de

datos relacionados a la combinación de las técnicas de control de calidad estáticas y dinámicas. Dicho estudio, realizara un mapeo sistemático contando con varios criterios de selección de datos. Por ejemplo, se seleccionara libros, revistas, entre otras fuentes de información segura, para categorizarlos dependiendo de su año de publicación, autor(es), nombre, idioma en el cual fue redactado, evidencia documentada, objetivos y conceptos claros (FRANK ELBERZHAGER JÜRGEN MÜNCH; VI TRAN NGOC NHA, 2012).

Al principio de la investigación se fue recopilando una gran cantidad de información, la cual fue descartada mediante los criterios previamente establecidos. Además, comenzaron a surgir nuevos criterios que se convirtieron en categorías de agrupamiento. Por ejemplo, se analizó que existían tipos de combinación de técnicas de control de calidad dinámico y estático. Sin embargo, se tomó mayor consideración a 2 de estos tipos, los cuales son compilación e integración. Mediante estas categorías, se comenzó a investigar y dividirse en subcategorías, las cuales fueron recopilación, combinación y pruebas. Siguiendo este patrón de búsqueda mediante categorías, dividiéndolos en subcategorías, asignándoles porcentajes de importancia y gráficamente se ha podido obtener información más viable para sustentar la combinación de estas técnicas (FRANK ELBERZHAGER JÜRGEN MÜNCH; VI TRAN NGOC NHA, 2012).

En conclusión, se ha podido obtener, analizar y comprobar que es óptimo la combinación de las técnicas de control de calidad dinámica y estática. Mediante este análisis, se demuestra que existirá una reducción de los tiempos de desarrollo, mayor eficiencia y eficacia. Además, con la información recopilada se demuestra, por ejemplo, que desde el año 2005 se ha comenzado a proponer ideas de combinación de técnicas. Por otro lado, más del 50% de los artículos analizados demuestran que esta propuesta ya

había comenzado a proponerse. Por lo tanto, existe una base contundente y concreta de precedentes que apoyan esta propuesta.

### 3.3.6 **Assessment of software quality: Choquet Integral Approach**

La medición de software es una disciplina que lidia con la cuantificación de las características de un producto software o un proceso de software. Existen muchos modelos de calidad de software, sin embargo, ninguno ofrece una cuantificación para los parámetros de calidad con los que son evaluados. Dado que calidad es un término relativo que difiere de los distintos puntos de vista (trascendental, manufactura, usuario y valor), muchos modelos de calidad fueron desarrollados con el propósito de producir aplicaciones software genéricas. Debido a ello, se hizo un esfuerzo en evaluar alternativas de solución proponiendo así un Software de Especificación de Requerimientos común basado en atributos de calidad que son altamente interdependientes. Estos atributos de calidad basados en varias perspectivas no son independientes las unas de las otras y son algo confusas en naturaleza. Es por ello que se planteó la Integral de Choquet dado que provee una gran flexibilidad y soporta muchos criterios de decisión (VATESH PASRIJA, SANJAY KUMAR, PRAVEEN RANJAN SRIVASTAVA, 2012).

- Descripción del aporte
  - 1- Elegir los criterios y su escala de importancia.
  - 2- Ingresar el valor de cada criterio según la alternativa.
  - 3- Encontrar las medidas inexactas.
  - 4- Dar un valor a cada alternativa.
  - 5- Calcular el peso de cada punto de vista.
  - 6- Comparar el valor del resultado de las alternativas.
- Métricas que el autor usa y resultado que obtiene.

Punto de vista del Usuario:

1. Confiabilidad.
2. Usabilidad.

Punto de vista del Producto:

3. Portabilidad.
4. Eficiencia.

Punto de vista de la Manufactura:

5. Flexibilidad.
6. Reusabilidad.

### **3.3.7 QRP: a CMMI Appraisal Tool for Project Quality Management**

Hoy en día las organizaciones buscan utilizar SPI (Software Process Improvement) Frameworks con el fin de encontrar una mayor eficacia en el proceso de desarrollo y mejor calidad del producto software final. CMMI se presenta como uno de estos frameworks, el cual, presenta una guía para mejorar el ciclo de vida del software, así como, permite evaluar los procesos de desarrollo; sin embargo, análisis identifican que las empresas buscan una herramienta que requiera el menor costo y tiempo posible. Frente a esta situación, se presenta a QRP (Quality Ready Portal) una plataforma que apoya el control de calidad durante las fases de desarrollo del proyecto, y que presenta como una de sus principales características la automatización de la evaluación del grado de cumplimiento de una CMMI maturity o el nivel de capacidad mediante el uso natural de la plataforma (ALLUÉ, ALBERTO; DOMÍNGUEZ, ELADIO; LÓPEZ, ANTONIO; ZAPATA, MARÍA A., 2013).

CMMI for Development (CMMI-DEV) es una recopilación de las mejores prácticas para la mejora de la ingeniería de procesos y desarrollo en las empresas que desarrollan productos y servicios. Con el objetivo de

identificar las fortalezas y debilidades de los procesos de una organización y la evaluación de la proximidad de la organización con las mejores prácticas CMMI, se llevan a cabo evaluaciones periódicas de conformidad con el modelo. En concreto, the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) es el método oficial de valoración CMMI que se utiliza para evaluar los procesos de las organizaciones y para proporcionar clasificaciones (ALLUÉ, ALBERTO; DOMÍNGUEZ, ELADIO; LÓPEZ, ANTONIO; ZAPATA, MARÍA A., 2013).

La propuesta ideal como herramienta es QRP debido a que integra el trabajo diario además de orientar y evaluar la calidad con un nivel CMMI. QRP ha sido diseñado con una arquitectura modular basada en tres niveles de generalidad: los niveles del sistema, de organización y de proyectos. La principal innovación a nivel de producto, la posibilidad de generar, rápida y semi-automática, la base de datos de evidencias objetivas que es requerido por las auditorías de cumplimiento. Las descripciones de los procesos se han agrupado de acuerdo con el nivel de generalidad al que pertenecen, se divide en 2 grupos:

- Nivel organizacional: el módulo de guía CMMI apoya la toma de decisiones en relación con el nivel y tipo de representación decisión (etapas o continuo) del CMMI-DEV que se va a implementar. Una vez tomada la decisión, las áreas de proceso, metas y prácticas que intervienen en el nivel CMMI que deben alcanzarse son establecidos por el módulo de implementación de CMMI.
- Nivel de Proyecto: El Proyecto de Procesos módulo permite al usuario adaptar el mapa para las necesidades del proyecto. Durante el trabajo diario de los proyectos, el módulo gestor de documentos permitirá la grabación de los diferentes artefactos documentales que se generan. El módulo de Business Intelligence permite definir indicadores de



desempeño para mediciones de procesos a nivel de proyecto, por ejemplo, el tiempo promedio para terminar un artefacto documental o la desviación del cronograma previsto. Este módulo también permite la realización de análisis de datos mediante cubos OLAP.

El módulo CMMI Appraisal implementa las normas que se establezcan por el

método SCAMPI para identificar las fortalezas y debilidades de los procesos de una organización y evaluar su proximidad a las mejores prácticas de CMMI.

El módulo CMMI Appraisal requiere el conjunto de selección de proyectos. Una vez que se seleccionan, QRP detecta automáticamente, como evidencia objetiva posible, todos los artefactos documentales que se registraron durante la ejecución de los proyectos seleccionados. Luego, las evidencias proporcionadas deben ser examinadas y validadas. Después, el módulo proporciona automáticamente una evaluación para cada una de las prácticas y las áreas de metas y procesos que corresponden al nivel evaluado. Por último, el informe con los resultados de la evaluación y la lista de evidencias objetivas previstas en cada una de las prácticas puede ser generado como un archivo de Excel (ALLUÉ, ALBERTO; DOMÍNGUEZ, ELADIO; LÓPEZ, ANTONIO; ZAPATA, MARÍA A., 2013).

En el presente Capítulo III se realizó una descripción de los principales modelos relacionados con la Calidad de Software y el proceso de Verificación y Validación de Software y se evidenciaron las principales desventajas con respecto a su aplicación en una pequeña empresa dedicada a la fabricación de software, en el siguiente Capítulo IV se mostrará el Marco Metodológico propuesto del proceso de Verificación y Validación de Software.

## **Capítulo 4: Marco Metodológico propuesto del Proceso de V&V de Software para las PYMES en el Perú**

En el presente capítulo se propone un nuevo Marco Metodológico del proceso de Verificación y Validación de Software para Pymes. Primero se describen los modelos de calidad que sirvieron como base para la propuesta, después se muestran los roles utilizados. En el siguiente punto se describe las cuatro fases propuestas con sus respectivas entradas y salidas, finalmente se describen las métricas y plantillas propuestas.

En primer lugar debemos de considerar que el proceso de Verificación y Validación de Software, es un proceso que se debe iniciar con el nacimiento de los proyectos o requerimientos y culminar en la puesta en marcha o pase a producción respectivo; la Verificación y Validación del Software no debe ser un proceso aislado que se activa al término de la construcción del Software, sino que debe estar integrado con las necesidades del negocio e ir de la mano con todo el proceso de Desarrollo.

### **4.1 Roles del proceso de Verificación y Validación de Software**

En esta sección tomaremos como referencia el acápite de Roles relacionado a TMap, seleccionando los roles básicos y prioritarios que una Pyme necesitaría:

- Líder del equipo de pruebas e inspecciones.
- Inspector.
- Analista de pruebas.
- Ejecutor de Pruebas (Testers).

En una realidad particular una sola persona podría cubrir los cuatro roles, esto dependerá del tamaño y los recursos con los que cuenta la Pyme. En la tabla 4.2 se muestra las funciones de cada rol.

Rol	Función
<b>Líder del equipo de pruebas e inspecciones</b>	<ol style="list-style-type: none"> <li>1. Será el responsable de las tareas de gestión, supervisión y control de los proyectos de V&amp;V de las aplicaciones asignadas. Las labores fundamentales se describen a continuación: <ul style="list-style-type: none"> <li>• Generación de los planes de V&amp;V, prueba, calendario de pruebas, plan de riesgos.</li> <li>• Coordinación de todas las actividades de V&amp;V, prueba, verificando y reportando la realización y aprobación de las actividades de V&amp;V.</li> <li>• Asegurar la ejecución de las inspecciones/pruebas en las fechas comprometidas.</li> <li>• Controlar la calidad y resultados finales de V&amp;V.</li> <li>• Administrar la relación con el cliente con respecto a las actividades de V&amp;V.</li> <li>• Asegurar el desarrollo de las actividades de V&amp;V oportunamente, siguiendo los estándares de inspecciones y pruebas.</li> <li>• Responsabilidad de gestión de V&amp;V de las aplicaciones y de la asignación de tareas para garantizar la consecución de los hitos que se ofertan. Se encargará de marcar las directrices del proyecto, controlar y gestionar la planificación del mismo, identificar y analizar las posibles desviaciones de la planificación definida, solucionar y/o escalar cualquier problema existente en el proyecto y gestionar riesgos.</li> </ul> </li> <li>2. Conocimientos en Técnicas de revisiones, inspecciones y diseño de casos de pruebas.</li> <li>3. Se recomienda estar certificado en ISTQB nivel avanzado.</li> </ol>
<b>Inspector</b>	<ol style="list-style-type: none"> <li>1. Persona con perfil técnico/funcional responsable de la elaboración de las listas de verificación y log de inspección, así como la ejecución y supervisión de las revisiones. Se encargarán de: <ul style="list-style-type: none"> <li>• Crear las listas de verificación y log de inspección para el proyecto.</li> <li>• Control de documentación verificada.</li> <li>• Labores de seguimiento hasta levantar todas las observaciones encontradas.</li> <li>• Elaboración de informes.</li> <li>• Desarrollo de acciones preventivas.</li> </ul> </li> <li>2. Conocimientos en Técnicas de revisiones e inspecciones.</li> <li>3. Se recomienda estar certificado en ISTQB.</li> </ol>

Rol	Función
<b>Analista de Pruebas</b>	<ol style="list-style-type: none"> <li>1. Persona con perfil técnico/funcional responsable de la elaboración de los diseños funcionales, así como del diseño, ejecución y supervisión de las pruebas funcionales de los productos desarrollados en mantenimiento evolutivo. Se encargarán de: <ul style="list-style-type: none"> <li>• Crear especificaciones de prueba para el proyecto, las cuales incluyen casos de prueba y datos de prueba para el proyecto.</li> <li>• Elaboración y ejecución de los planes de prueba.</li> <li>• Ejecución de pruebas funcionales.</li> <li>• Control de Operativa Diaria.</li> <li>• Labores de seguimiento del reléase probado: planificación, obtención de documentación, seguimiento de entregas y realización de pruebas.</li> <li>• Elaboración de informes.</li> <li>• Desarrollo de acciones preventivas.</li> <li>• Documentación de las evidencias de pruebas.</li> </ul> </li> <li>2. Conocimientos en Técnicas de diseño de casos de pruebas.</li> <li>3. Se recomienda estar certificado en ISTQB.</li> </ol>
<b>Ejecutor de Pruebas (Tester)</b>	<ol style="list-style-type: none"> <li>1. Responsable de garantizar la implementación técnica de las tareas de prueba necesarias. Desde un punto de vista general, las tareas asumidas por este perfil serán: <ul style="list-style-type: none"> <li>• Ejecutar casos de pruebas como fue definido en el plan de prueba, bajo la dirección del analista de pruebas.</li> <li>• Documentar y reportar los resultados de la ejecución de pruebas.</li> <li>• Monitorización y operativa de la ejecución de pruebas.</li> <li>• Seguimiento de la Incidencias.</li> </ul> </li> <li>2. Se recomienda estar certificado en ISTQB.</li> </ol>

Tabla 4-1 Roles de Verificación y Validación (Elaboración Propia)

## 4.2 Responsabilidades de las áreas que participan en el proceso de Verificación y Validación de Software

A continuación se detalla las responsabilidades a seguir por área. Consideramos que una Pyme podría tercerizar el área de desarrollo y de V&V, pero deberá dejar sentado cuales son las responsabilidades de cada una, las cuales pasamos a listar:

### 1. Negocio

- Levantamiento inicial de requerimientos de los usuarios.
- Establecimiento de reuniones para entendimiento de la solicitud del usuario -> Negocio / Desarrollo / Calidad.
- Elaboración de la especificación funcional detallada.

## 2. Comité

- Planificación y Priorización de las atenciones teniendo en cuenta los objetivos y metas de la organización.

## 3. Oficina de Desarrollo

- Análisis inicial de la solicitud del usuario.
- Análisis de Impacto.
- Elaboración de la estimación del esfuerzo y tiempo de atención.
- Elaboración de la Especificación técnica detallada.
- Manual de Configuración e Instalación.
- Construcción de la solución.
- Soporte durante el proceso de pruebas de calidad.
- Elaboración del plan de Reversión

## 4. Oficina de Verificación y Validación

- Aseguramiento de la alineación y trazabilidad de la solicitud inicial del usuario con los documentos de especificación y construcción de la solución.
- Elaboración de la lista de Verificación Funcional y Propuesta técnica inicial.
- Elaboración de la estimación del esfuerzo y tiempo de atención.
- Establecimiento del Plan de Pruebas de la atención.
- Elaboración del Listas de verificación para la Especificación detallada Funcional y Técnica.
- Diseño de los casos y escenarios de pruebas.
- Consolidado de las evidencias y reporte de defectos.
- Establecimiento de Métricas y Elaboración de informes de avance.

### **4.3 Descripción del Marco Metodológico Propuesto para el proceso de Verificación y Validación de Software**

En el esquema Figura 3-1 se muestra un proceso cíclico e iterativo que nace con las necesidades de los usuarios o identificación de requerimientos hasta la puesta en producción del software. El Marco considera actividades de Planeación, Análisis, Diseño, Pruebas y Ratificación (es el nombre utilizado para las pruebas que realiza el usuario en un ambiente controlado o también denominado ambiente pre-productivo). Por otro lado, también se tomará en cuenta las métricas y los procesos propuestos según los modelos y estándares del proceso Verificación y Validación de Software más representativos del mercado.

Considerando las buenas prácticas el proceso de Verificación y Validación de software se realizará en 2 niveles:

- V&V Estática: Relacionada con el análisis de una representación estática de un sistema para descubrir algún problema, puede ser documento, un diagrama o incluso pseudocódigo. El Marco Propuesto considera desde la verificación y validación estática de la solicitud inicial presentado por el negocio hasta la revisión de la especificación técnica.
- V&V Dinámica: Relacionadas con las pruebas y observaciones del comportamiento real del software. El sistema es ejecutado con datos de prueba en un ambiente controlado y el tester observará el comportamiento operacional. El Marco propuesto considera sólo pruebas funcionales porque son las pruebas más utilizadas y las que deben realizarse como mínimo antes de sacar a producción un software.

Restricciones:

- El Marco propuesto no incluye pruebas automatizadas como pruebas seguridad, usabilidad o performance. Tampoco se incluye la pruebas unitarias o realizadas por el programador.

- El Marco propuesto está adaptado para ser consecuente con los procesos generales que mantienen las Pymes; sin embargo su implementación final dependerá de la realidad de cada una de ellas.

A continuación en la Figura 4-1 se muestra las fases del ciclo de vida del software y la relación con el proceso de Verificación y Validación de software. Lo que se busca es alinear los procesos de Verificación y Validación de Software a los procesos del negocio y de Desarrollo de Software.

De manera horizontal se muestran la cinco fases comunes: Planeación, Análisis, Diseño, Pruebas y Ratificación. El proceso de Verificación y Validación estática se inicia desde la planeación hasta el diseño y el proceso de Verificación y Validación dinámica abarca desde las pruebas hasta la ratificación. Durante todo el proceso de Verificación y Validación se detallarán las métricas más importantes que permitirán realizar ajustes al proceso inicial.

El Marco es flexible y funcionaría si el desarrollo lo realiza un ente externo a la organización o es una oficina dentro de la misma empresa.

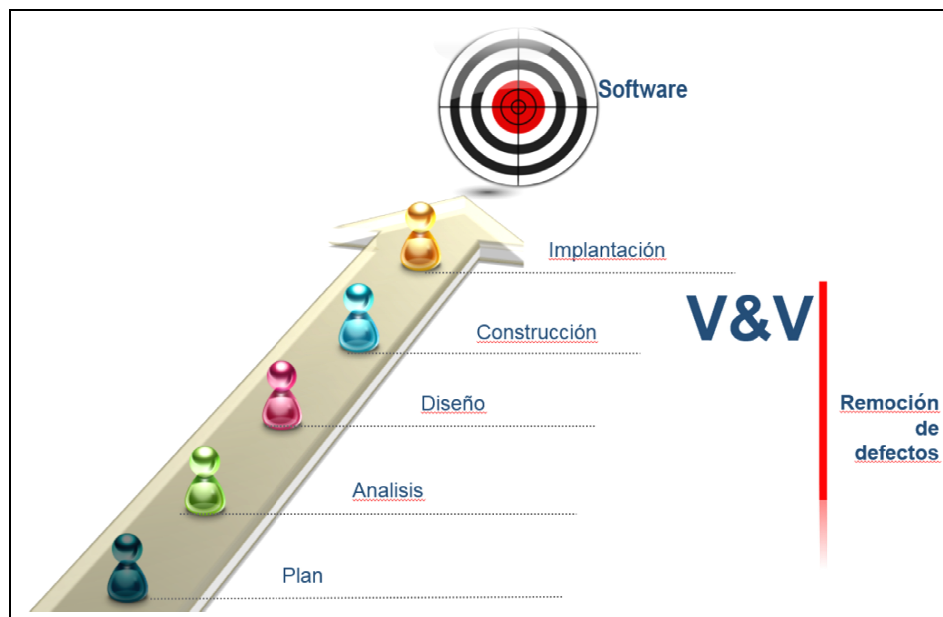


Figura 4-1 Esquema General: Fases del ciclo de vida del SW Vs. Proceso V&V (Elaboración Propia)

El Marco Propuesto del proceso de Verificación y Validación consta de cuatro etapas: Planeación, Análisis, Diseño y Pruebas.

- En el caso de la **Fase de Planeación del Proceso de Verificación y Validación de Software** se alinea con las actividades ejecutadas por el área de Desarrollo: Construcción de la Especificación Funcional, Especificación Técnica, estimación de la solución, priorización y planificación.
- La **Fase de Análisis del Proceso de Verificación y Validación de Software** se alinea con las actividades que permiten el diseño Funcional y técnico.
- La **Fase de Diseño del Proceso de Verificación y Validación de Software** se alinea con las actividades que permiten la construcción y pruebas.
- Finalmente la **Fase de Pruebas/Ratificación** se alinea se alinea con las actividades de Soporte y Post Producción.

A continuación se describen las entidades que participan en el proceso de Verificación y Validación: El Comité de negocio, el área de desarrollo y el área de Calidad:

- **Comité de Negocio:** el primer punto es entregar una Solicitud Funcional o plasmar en un documento la necesidad inicial, dependiendo de las necesidades el Comité de Negocios deberá realizar una priorización y aprobar la planificación del Proyecto o requerimiento. En algunas Pymes el Comité de Negocio puede generar la Especificación Funcional. Para el Marco propuesto se ha considerado que esta actividad es parte de sus responsabilidades. En otras Pymes, el Comité de Negocios puede listar los requerimientos y el área de desarrollo sería el encargado de la generación de este documento. Independientemente de qué área genere el documento de Especificación Funcional, el área de calidad siempre se encargará de utilizar los procesos de Verificación y Validación para asegurar que esta solicitud está alineada con lo que el negocio requiere.
- **Oficina de Desarrollo:** Deben entregar una estimación de tiempo y costo en base a la Solicitud Inicial. Después de que la Especificación Funcional ha sido



verificada deberán generar la Especificación Técnica. Los entregables principales son el código fuente, los manuales respectivos de instalación, configuración, usuario y plan de reversión o Rollback.

- **Oficina de Calidad- V&V:** La oficina de calidad que participará en los procesos de Verificación y Validación de Software, deberá revisar la solicitud inicial y entregar un estimado de tiempo y costo a alto nivel. Como los requerimientos están a muy alto nivel, en el Marco se propone una forma que apoye con la estimación inicial. Después de que el Comité aprobó el requerimiento se trabajará en la revisión de la documentación: sólo especificación funcional y técnica. Se entregará el plan detallado incluyendo la elaboración de casos de prueba. Cuando el software ha culminado su construcción, se iniciarán las pruebas funcionales, los entregables serán el reporte de defectos y las evidencias. Finalmente se entregarán los indicadores de Gestión. A continuación se describen cada una de las Fases del Proceso de Verificación y Validación de Software con sus respectivas entradas, salidas, roles que intervienen en el proceso y una descripción de iniciativas.

La figura 4-2 muestra el Marco Propuesto con las cuatro fases del proceso de Verificación y Validación: Planeación, Análisis, Diseño y Pruebas y los entregables de cada una de las fases:



Figura 4-2 Fases del Marco Propuesto para el Proceso de V&V de Software (Elaboración Propia)

#### 4.3.1 Fase de Planeación del Proceso de Verificación y Validación de Software

Esta Fase se divide en dos partes que permiten entregar una Solicitud Formal Verificada y una Estimación a alto nivel.

##### 4.3.1.1 Fase de Planeación del Proceso de Verificación y Validación de Software: Solicitud Formal Verificada

El Marco Metodológico Propuesto del Proceso de Verificación y Validación de Software se activa con la entrega de una Solicitud Funcional, este documento es generado por el Comité del Negocio y será el gatillo principal para activar los procesos de Desarrollo y Calidad (de manera específica el proceso de Verificación y Validación).

Los equipos de Desarrollo y Calidad revisaran la Solicitud Funcional, este documento contiene los requerimientos del usuario y debe ser claro. A pesar de que en esta etapa del Proyecto los requerimientos no tienen el suficiente detalle,

debemos comunicar al usuario los riesgos y problemas que se pueden ocasionar en la estimación. Es importante que el equipo de Calidad convoque a reunión a los equipos de Desarrollo y Comité de Negocios con la finalidad de asegurar el entendimiento de los requerimientos iniciales, de esta manera la estimación será más certero.

A continuación, en la Figura 4-3 se muestra el Diagrama de la Fase de Planeación del Proceso de Verificación y Validación de Software.

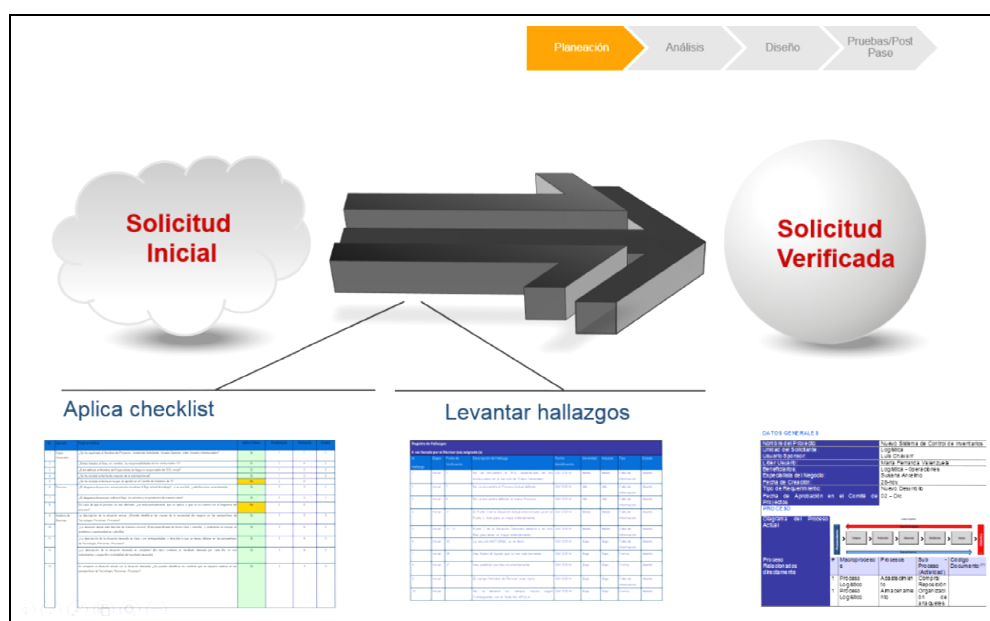


Figura 4-3 Fase de Planeación del Proceso de V&V de SW: Solicitud Inicial (Elaboración Propia)

En la Tabla 4-3 se muestra las entradas, salidas, roles que intervienen en el proceso y una descripción de iniciativas para generar la Solicitud Formal Verificada.

Ítem	Descripción
Entrada	Solicitud Funcional
Salida	Solicitud Formal Verificada

<b>Rol</b>	Líder del equipo de Inspecciones y Pruebas
<b>Iniciativa</b>	Esta <b>Solicitud Formal</b> , deberá pasar por un proceso de <b>Inspección y Revisión de Solicitud Funcional</b> utilizando una lista de verificación o Listas de verificación previamente elaborado y aprobado por el negocio.
<b>Otros Entregables</b>	<ul style="list-style-type: none"> <li>• Reportes de Hallazgos</li> <li>• Indicadores</li> </ul>
<b>Estándares relacionados</b>	<ul style="list-style-type: none"> <li>• IEEE Std 730-2002</li> <li>• IEEE Std 829-1998</li> <li>• IEEE Std 1028-1997</li> <li>• IEEE Std 1012-1997</li> <li>• IEEE Std 610.12-1990</li> </ul>

Tabla 4-2 Fase de Planeación del Proceso de V&V de SW: Solicitud Inicial (Elaboración Propia)

#### 4.3.1.2 Fase de Planeación del Proceso de Verificación y Validación de Software: Estimación Inicial

Con Solicitud Formal Verificada los equipos de Desarrollo y Calidad podrán realizar las estimaciones de tiempo y costo a alto nivel. El equipo de desarrollo elabora el documento de Estimación, Análisis de Impacto, Cronograma y el Equipo de Calidad elabora el documento de Identificación de alcance de V&V y cronograma de alto nivel. A continuación, en la Figura 4-4 se muestra el Diagrama de la Fase de Planeación del Proceso de Verificación y Validación de Software.



<b>relacionados</b>	<ul style="list-style-type: none"> <li>• IEEE Std 829-1998</li> <li>• IEEE Std 1028-1997</li> <li>• IEEE Std 1012-1997</li> <li>• IEEE Std 610.12-1990</li> </ul>
---------------------	---

Tabla 4-3 Fase de Planeación del Proceso de V&V de SW: Estimación Inicial (Elaboración Propia)

### **4.3.2 Fase de Análisis del Proceso de Verificación y Validación de Software**

Esta Fase se divide en dos partes que permiten entregar una Especificación Funcional Verificada y una Plan de Verificación y Validación.

#### **4.3.2.1 Fase de Análisis del Proceso de Verificación y Validación de Software: Especificación Funcional/Técnica Verificada**

Después de obtener la aprobación de la Solicitud Funcional, se iniciará la creación de la Especificación Funcional, este documento será generado por el negocio. La especificación funcional pasará por un proceso de Verificación y Validación de Software que permita asegurar que la Especificación Funcional está alineada a los requerimientos iniciales del usuario.

El objetivo principal del equipo es detectar la mayor cantidad de defectos, los defectos serán listados en un Log de Inspección para que puedan ser levantados por el área respectiva. Este proceso es cíclico y se repetirá tantas veces como sea necesario hasta levantar todos los defectos encontrados según las políticas de la empresa.

A continuación, en la Figura 4-5 se muestra la Fase de Análisis del Proceso de V&V de SW: EEFF/EETT.

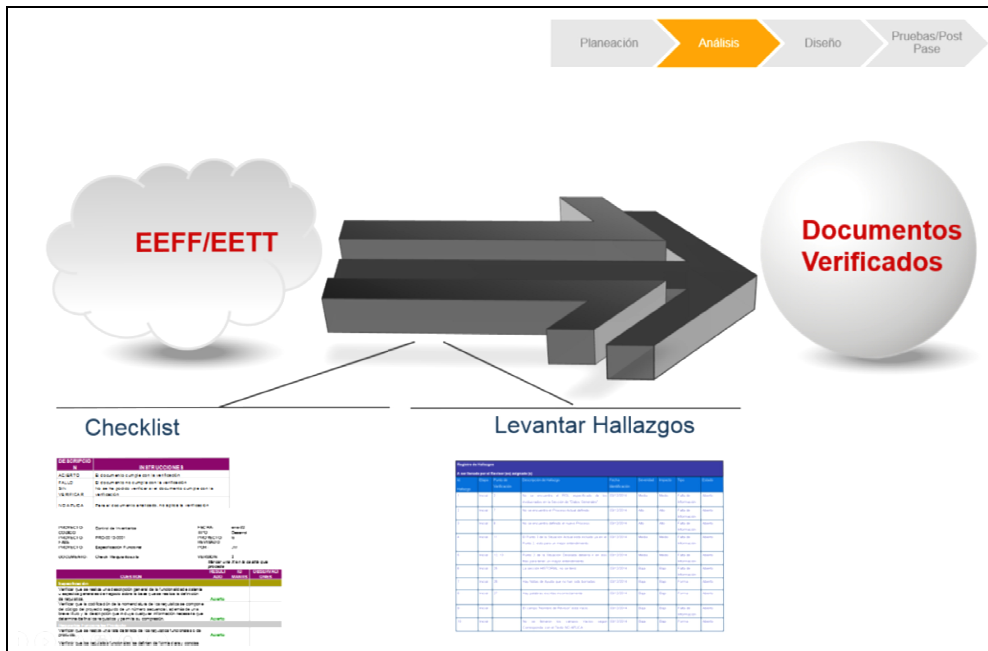


Figura 4-5 Fase de Análisis del Proceso de V&V de SW: EEFF/EETT (Elaboración Propia)

En la Tabla 4-5 se muestra las entradas, salidas, roles que intervienen en el proceso y una descripción de iniciativas para generar la Especificación Funcional Verificada.

Item	Descripción
<b>Entrada</b>	<b>Especificación Funcional</b> <b>Identificación de alcance de V&amp;V y cronograma de alto nivel.</b>
<b>Salida</b>	<b>Especificación Funcional Verificada</b>
<b>Rol</b>	Inspectores
<b>Iniciativa</b>	El equipo de inspectores deberá revisar la especificación funcional y se apoyará en un log de inspección para listar los defectos encontrados. Este log de inspección deberá ser entregado al área que generó la especificación funcional para que pueda levantar los defectos. El documento corregido volverá a ingresar nuevamente para que el equipo de inspectores revise si los defectos han sido levantados correctamente. Este proceso es cíclico y se repite de acuerdo a las políticas de la empresa

<b>Otros Entregables</b>	Ninguno
<b>Estándares relacionados</b>	<ul style="list-style-type: none"> <li>• IEEE Std 730-2002</li> <li>• IEEE Std 829-1998</li> <li>• IEEE Std 1028-1997</li> <li>• IEEE Std 1012-1997</li> <li>• IEEE Std 610.12-1990</li> </ul>

Tabla 4-4 Fase de Análisis del Proceso de V&V de SW: EEFF (Elaboración Propia)

Después de obtener una Especificación Funcional consistente con los requerimientos, el equipo de desarrollo elabora la Especificación Técnica, este documento también pasará por un proceso de Verificación y Validación. Paralelamente a la elaboración de la Especificación Técnica el equipo de Calidad iniciará con la elaboración del Plan de Verificación y Validación/Pruebas de Software.

#### **4.3.2.2 Fase de Análisis del Proceso de Verificación y Validación de Software: Plan de Verificación y Validación**

La Especificación Funcional verificada es el input principal para la generación del Plan del Proyecto y el Plan de Verificación y Validación. El Plan de Verificación y Validación incluye el Plan de Pruebas. Como se ha mencionado anteriormente, el Marco Metodológico propuesto sólo incluye pruebas funcionales, por ello, el Plan de Pruebas sólo incluirá las actividades relacionadas al diseño y ejecución de casos de pruebas funcionales debido a que este tipo de pruebas es el más solicitado en las empresas y no necesariamente se requiere una herramienta de automatización porque se puede realizar en forma manual.

En la Tabla 4-6 se muestra las entradas, salidas, roles que intervienen en el proceso y una descripción de iniciativas para generar el Plan de Verificación y Validación de Software.

<b>Item</b>	<b>Descripción</b>
<b>Entrada</b>	<b>Especificación Funcional Verificada</b>



	<b>Identificación de alcance de V&amp;V y cronograma de alto nivel.</b>
<b>Salida</b>	<b>Plan de Verificación y Validación</b>
<b>Rol</b>	Líder del equipo de Inspecciones y Pruebas
<b>Iniciativa</b>	En base a los documentos Identificación de alcance de V&V, cronograma de alto nivel y la especificación funcional, se creará el plan de V&V
<b>Otros Entregables</b>	Ninguno
<b>Estándares relacionados</b>	<ul style="list-style-type: none"> <li>• IEEE Std 730-2002</li> <li>• IEEE Std 829-1998</li> <li>• IEEE Std 1028-1997</li> <li>• IEEE Std 1012-1997</li> <li>• IEEE Std 610.12-1990</li> </ul>

Tabla 4-5 Fase de Análisis del Proceso de V&V de SW: Plan de V&V (Elaboración Propia)

#### 4.3.3 Fase de Diseño del Proceso de Verificación y Validación de Software

Si el Plan de Verificación y Validación de Software es aprobado, se iniciará con el proceso de diseño de pruebas; si no es aprobado tendrá que ajustarse hasta que el negocio lo apruebe. Los documentos de Especificación Funcional y Técnica verificados serán la entrada principal para elaborar los escenarios y casos de pruebas, los cuales también deberán ser aprobados por el negocio. Paralelamente las especificaciones funcionales y técnicas pasarán al área de desarrollo para que se inicie el proceso de Construcción de Software.

A continuación, en la Figura 4-6 se muestra Fase de Diseño del Proceso de V&V de SW: Casos de Prueba.

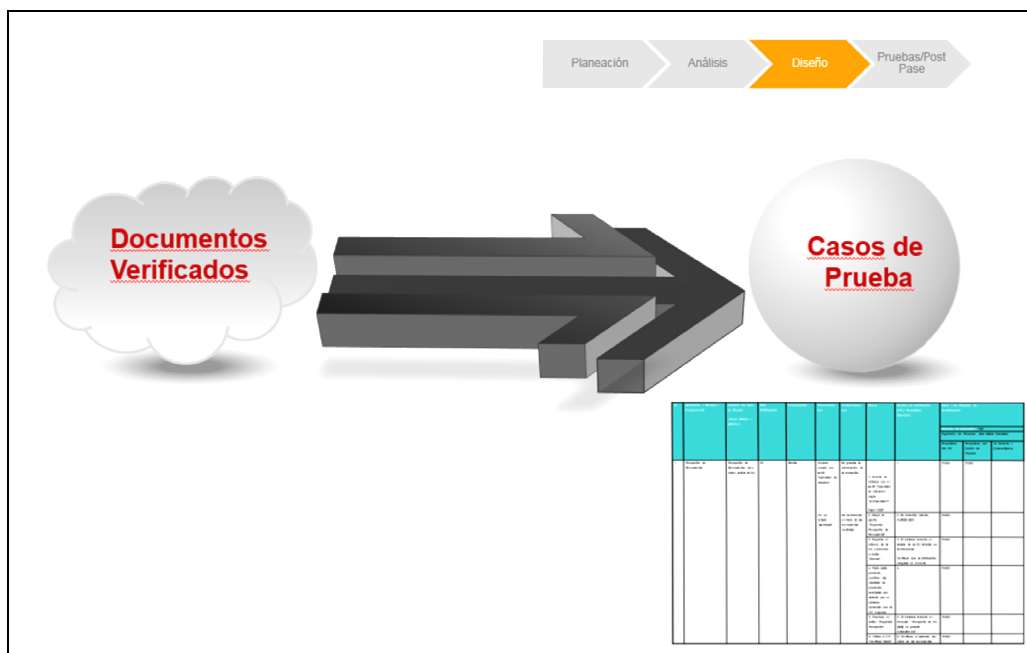


Figura 4-6 Fase de Diseño del Proceso de V&V de SW: Casos de Prueba (Elaboración Propia)

En la Tabla 4-7 se muestra las entradas, salidas, roles que intervienen en el proceso y una descripción de iniciativas para generar los entregables del Diseño de Pruebas.

Item	Descripción
<b>Entrada</b>	<b>Especificación Funcional Verificada</b> <b>Especificación Técnica Verificada</b> <b>Plan de V&amp;V</b>
<b>Salida</b>	Escenarios de Pruebas Casos de Pruebas
<b>Rol</b>	Analistas de Pruebas
<b>Iniciativa</b>	El equipo de analistas de Pruebas iniciará con la elaboración de los escenarios de pruebas que serán revisados con el negocio hasta obtener su aprobación. Después de obtener la aprobación de los escenarios de pruebas, los analistas iniciarán con la elaboración de los casos de pruebas detallados.

<b>Otros Entregables</b>	Ninguno
<b>Estándares relacionados</b>	<ul style="list-style-type: none"> <li>• IEEE Std 730-2002</li> <li>• IEEE Std 829-1998</li> <li>• IEEE Std 1028-1997</li> <li>• IEEE Std 1012-1997</li> <li>• IEEE Std 610.12-1990</li> </ul>

Tabla 4-6 Fase de Diseño del Proceso de V&V de SW: Diseño de Pruebas (Elaboración Propia)

#### **4.3.4 Fase de Pruebas y Ratificación del Proceso de Verificación y Validación de Software**

Después de que el equipo de Desarrollo termine la construcción del Software se iniciará el proceso iterativo de ejecución de pruebas. El área de desarrollo debe entregar una versión congelada del Software y debe instalarla en el ambiente de pruebas. Adicionalmente, debe concluir las pruebas unitarias y entregar las evidencias de sus pruebas. Con la versión instalada y los entregables recibidos, el equipo de pruebas iniciará el proceso de ejecución de pruebas. Este proceso también es iterativo y finalizará cuando todas las incidencias se han levantado o cuando el negocio lo solicite. Una vez que los criterios de calidad se hayan alcanzado el negocio podrá proceder con la puesta en marcha de la solución.

A continuación, en la Figura 4-7 se muestra Fase de Pruebas/Ratificación del Proceso de V&V de SW.

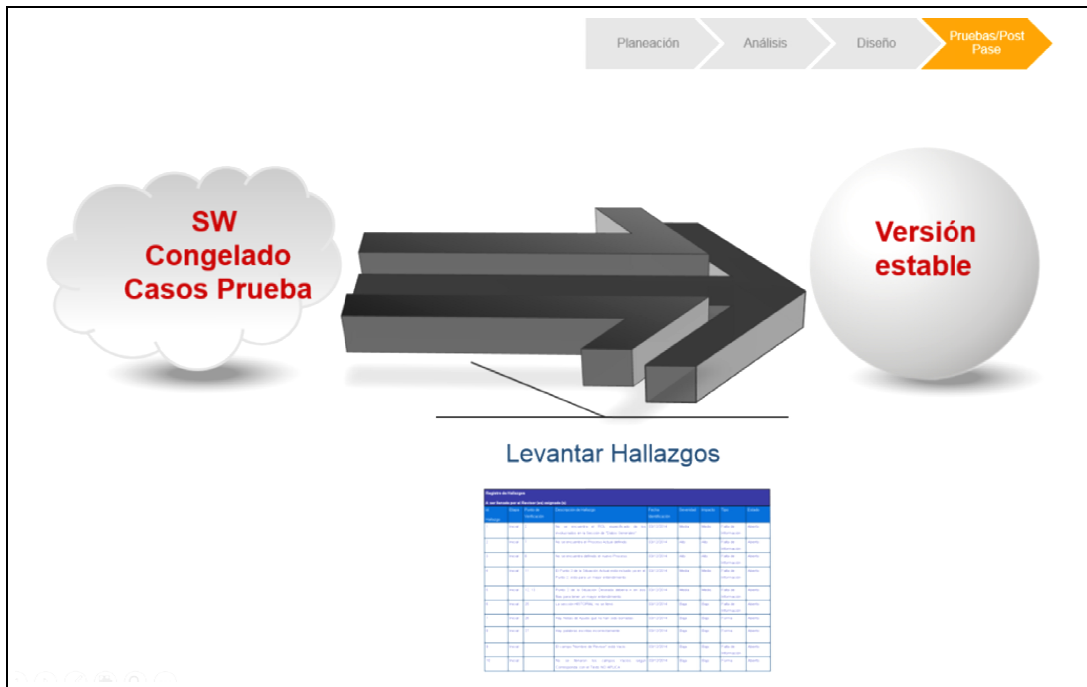


Figura 4-7 Fase de Pruebas/Ratificación del Proceso de V&V de SW (Elaboración Propia)

En la Tabla 4-8 se muestra las entradas, salidas, roles que intervienen en el proceso y una descripción de iniciativas para generar las evidencias de la Fase de Pruebas y Ratificación.

Ítem	Descripción
<b>Entrada</b>	<b>Plan de Pruebas</b> <b>Casos de Pruebas</b> <b>Versión Congelada del Software</b>
<b>Salida</b>	Reporte de defectos Evidencias de la ejecución de pruebas
<b>Rol</b>	Líder del equipo de Inspecciones y Pruebas Ejecutor de Pruebas
<b>Iniciativa</b>	El equipo de pruebas iniciará con la ejecución de los casos de pruebas. Los entregables de esta etapa son el reporte de defectos y los documentos de evidencia. Estos documentos deberán ser

	claramente especificados con la finalidad que el equipo de desarrollo pueda darle una pronta solución.
<b>Otros Entregables</b>	Ninguno
<b>Estándares relacionados</b>	<ul style="list-style-type: none"> <li>• IEEE Std 730-2002</li> <li>• IEEE Std 829-1998</li> <li>• IEEE Std 1028-1997</li> <li>• IEEE Std 1012-1997</li> <li>• IEEE Std 610.12-1990</li> </ul>

Tabla 4-7 Fase de Pruebas del Proceso de V&V de SW: Ejecución de Pruebas (Elaboración Propia)

#### 4.4 Gestión de Defectos para el Proceso de Verificación y Validación de Software

En los Procesos de Verificación y Validación estática y dinámica el equipo de Calidad detecta defectos o fallas que deben ser levantados o corregidos por sus respectivos autores, por ello, es importante una adecuada Gestión de Defectos.

En base a las Metodologías revisadas se propone una Gestión de defectos que permita planificar, revisar y hacer seguimiento de los mismos de una manera sencilla y ágil. Las actividades deben ser ejecutadas en el siguiente orden:

- **Identificar defectos:** analizar las listas de verificación o lista de inspección para identificar los defectos de la documentación. Este procedimiento se realiza cuando se ejecuta los procesos de V&V estática. Si se realiza V&V dinámica entonces se deberá identificar defectos comparando el resultado esperado de los casos de pruebas versus el resultado real.
- **Documentar defectos:** documentar el detalle de cada uno de los defectos identificados, incluyendo su prioridad, severidad y la fecha de compromiso para su solución.
- **Resolver los defectos:** llevar a cabo las acciones necesarias para que los autores resuelvan los defectos detectados por el equipo de calidad. Se debe

realizar un adecuado seguimiento de los defectos o incidencias hasta el cierre de los mismos, indicando la fecha de cierre y dejando constancia de dicho cierre.

- **Informar el estado de los defectos:** después de resolver los defectos encontrados se procede a informar el estado final de los mismos al responsable del Proyecto.

Los defectos registrados sirven como base de conocimiento para las lecciones aprendidas.

#### 4.5 Indicadores del Proceso de Verificación y Validación de Software

TMap proporciona indicadores útiles para hacer valoraciones sobre la calidad, se aplicarán algunas de ellas que se estimen básicas y mínimas para las Pymes.

A continuación, en la Figura 4-8 se muestra los indicadores utilizados en el Marco de V&V propuesto.

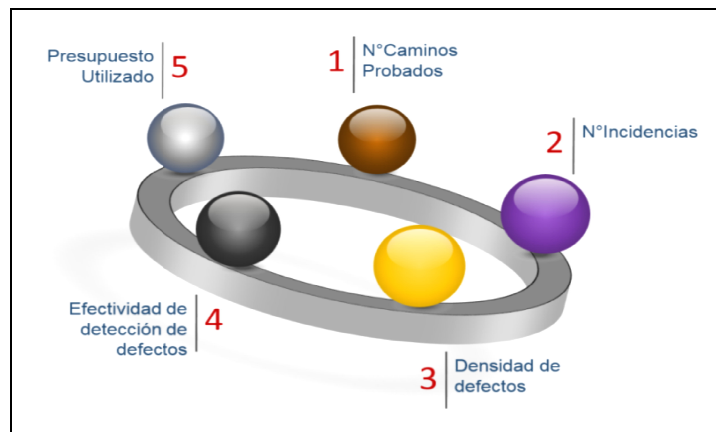


Figura 4-8 Indicadores utilizados en el Marco del proceso de V&V propuesto (Elaboración Propia)

- 1. Numero de caminos probados.** Proporción entre el número de rutas probadas y el total de rutas existentes. Esto permite medir porcentualmente la cantidad de rutas que el proceso de V&V considera probar a través de los casos de pruebas propuestos. La fórmula es la siguiente:

$$\text{Número de caminos probados} = \frac{\text{cantidad de caminos probados}}{\text{cantidad total de caminos}} \times 100\%$$

Donde:

- **Número de caminos probados:** es igual los flujos básico, alternativo y de excepción de los casos de uso probados. Es importante recordar que debido a las limitaciones de tiempo, costo y recursos las pruebas a ejecutar se delimitan a un sub conjunto de casos de pruebas priorizados.
- **Total de caminos:** es igual a la sumatoria de los flujos básico, alternativo y de excepción de cada una de las especificaciones de casos de uso definidos para el sistema a probar.

**2. Efectividad de la detección de defectos.** La métrica indica la efectividad del de las actividades de V&V para hallar defectos. Es el porcentaje obtenido de la cantidad de defectos hallados en las actividades de V&V y el total de defectos (defectos hallados en V&V + defectos hallados en producción). La fórmula es la siguiente:

$$= \frac{\text{defectos hallados en las pruebas}}{\text{defectos hallados en las pruebas} + \text{defectos hallados en producción}} \times 100\%$$

Donde:

- **Defectos hallados en las pruebas:** es igual a la cantidad de incidencias encontradas durante el proceso de pruebas
- **Defectos hallados en producción:** es igual a la cantidad de incidencias encontradas durante el uso real del software a probar.

**3. Presupuesto utilizado en revisiones.** Proporción del presupuesto y el costo actual de las pruebas. La métrica será utilizada para estimar el costo de implementar las actividades de V&V en un proyecto. La fórmula es la siguiente:

$$\text{Presupuesto Utilizado en Revisiones} = \frac{\text{presupuesto revisiones}}{\text{presupuesto total del proyecto}} \times 100\%$$

Donde:

- **Presupuesto de revisiones:** es igual al presupuesto utilizado para ejecutar el proceso de Verificación y Validación estático.
- **Presupuesto total del proyecto:** es igual al presupuesto utilizado para implementar todo el proyecto.

#### 4. Incidencias Por Tipo de Revisiones y Pruebas.

Esto permite medir la efectividad de las actividades de Verificación y Validación en cada tipo de revisión. La cantidad de defectos debe estar mapeada y debe ser compartida con el equipo de Desarrollo frecuentemente.

5. **Densidad de Defectos por Módulos.** La medición se basa en el principio de pruebas: “Agrupamiento de Defectos”. El principio define que un número pequeño de módulos contienen la mayoría de los defectos descubiertos durante el proceso de pruebas. Esta información se sustenta en las diferentes estadísticas y estudios realizados en la industria del software. IBM descubrió que el 75% de los errores se agrupaban en el 7% de los módulos (Jones). El 20% de los módulos de un programa son normalmente responsables del 80% de los errores (Boehm).

### 4.6 Plantillas del Proceso de Verificación y Validación de Software

#### 4.6.1 Planeación

##### 4.6.1.1 Planeación: Solicitud formal

De acuerdo al Marco Metodológico propuesto para el proceso de Verificación y Validación de software, se solicitará al negocio entregar el documento de solicitud inicial de requerimientos. Este documento deberá tener la siguiente información:



- Datos generales de la solicitud.
- Esquema de proceso actual y procesos relacionados.
- Descripción de la necesidad.
- Alineamiento estratégico para identificar la prioridad del proyecto dentro de la cartera de negocios.

En la Tabla 4-9 se muestra la plantilla de Solicitud Inicial.

## SOLICITUD - SECCIÓN INICIAL

### DATOS GENERALES

Nombre del Proyecto:	
Unidad del Solicitante:	
Usuario Sponsor:	
Líder Usuario:	
Beneficiarios:	
Especialista del Negocio:	
Fecha de Creación:	
Tipo de Requerimiento:	
Fecha de Aprobación en el Comité de Proyectos	

### PROCESO

Diagrama del Proceso Actual					
Proceso Relacionados directamente	#	Macroproceso	Procesos	Sub - Proceso (Actividad)	Código de Documento <sup>(1)</sup>
Proceso Relacionados indirectamente	Nº	Sub - Proceso (Actividad)	Descripción del Impacto		

	1					
	2					
<b>DESCRIPCIÓN DE LA NECESIDAD</b>						
Diagnóstico de la Situación Actual	1					
	2					
Resumen del Análisis Realizado	1					
	2					
<b>ALINEAMIENTO ESTRATÉGICO</b>						
Forma Parte de un Proyecto Estratégico de la Compañía						
Forma Parte de un Proyecto Estratégico de la División						
<b>Impacto / Alineamiento a los Pilares de Gestión</b>						
Pilar de Gestión	SI/NO	Justificación del alineamiento				

Tabla 4-8 Plantilla de Solicitud Inicial para la Estimación (Elaboración Propia)

Después de recibir el documento de Solicitud, el equipo de V&V realizará la verificación del documento utilizando un listas de verificación de Inspección. En la Tabla 3-10 se presenta la plantilla para las listas de verificación de Solicitud Inicial. Las instrucciones son las siguientes:

### 1. Identificar los puntos a verificar

- a. Deberá revisar los puntos a verificar y determinar cuáles son aplicables al producto que está revisando, en función del alcance del

mismo, en donde no aplique el criterio, deberá colocar “NO” en la columna “**Aplica Criterio**” y deberá registrar en una columna adicional las “Observaciones” que justifiquen esta exclusión. El resto de criterios deberá contener el texto “SI” para ser contabilizados en las métricas. La columna ponderación debe ser establecida por el negocio según el grado o nivel de importancia.

## **2. Revisar el producto de trabajo**

- a. Se debe calificar el nivel de cumplimiento del punto de verificación:
  - i. En la columna “**Ponderación**” se asignan valores basándose en la importancia del criterio a aplicar: tendrá 0 si no aplica, 1 si es indiferente, 2 si es importante y 3 si es muy importante.
  - ii. En la columna “**Estado**” tendrá el número 1 si no se cumplió el criterio o falta información; número 2 si se cumplió parcialmente, es decir hay hallazgos detectados y número 3 si se cumplió completamente el criterio. Después de llenar esta columna se calculará automáticamente la puntuación.
  - iii. La columna “**Puntuación**” puede tomar valores entre 0 y 10, siendo 0 el menor valor y 10 el mayor valor. La puntuación será asignada por el usuario de negocio.

## **3. Registrar los hallazgos**

- a. Registrar los hallazgos en el reporte de hallazgos, para cada uno de los criterios que tienen un estado con valor 1 o 2, deberán especificar el detalle del hallazgo correspondiente.

A continuación en la Tabla 4-10 se muestra la plantilla que deberá ser utilizada como Listas de verificación de Solicitud Inicial.

No.	Sección	Punto a Verificar	Aplica Criterio	Ponderación	Puntuación	Estado
1	<b>Datos Generales</b>	¿Se ha registrado el Nombre del Proyecto, Unidad del Solicitante, Usuario Sponsor, Líder Usuario e Involucrados?				
2		¿Están listadas el Área, rol, nombre, las responsabilidades de los involucrados?				
3		¿Está definido el Nombre del Especialista de Negocio responsable?				
4		¿Se ha incluido la fecha de creación de la solicitud Inicial?				
5		¿Se ha incluido la fecha en la que se aprobó en el Comité de Gobierno de TI?				
6	<b>Proceso</b>	¿El diagrama de proceso actual permite visualizar el flujo actual de trabajo?, si es una link, ¿esta funciona correctamente?.				
7		¿El diagrama de proceso indica el flujo, los actores y los productos de manera clara?				
8		En caso de que el proceso no sea afectado, ¿se está puntualizando que no aplica, o que no se cuenta con el diagrama del proceso?				
9	<b>Análisis de Brechas</b>	La descripción de la situación actual, ¿Permite identificar las causas de la necesidad del negocio en las perspectivas de Tecnología, Personas, Procesos?				
10		¿La situación actual está descrita de manera concisa? (Esta especificada de forma clara y sencilla; y solamente se incluye un problema u oportunidad en cada fila)				
11		¿La descripción de la situación deseada es clara y sin ambigüedades, y describe lo que se desea obtener en las perspectivas de Tecnología, Personas, Procesos?				
12		¿La descripción de la situación deseada es completa? (Es decir contiene un resultado deseado por cada fila, no son redundantes y especifica la totalidad del resultado deseado).				
13		Al comparar la situación actual con la situación deseada, ¿Se pueden identificar los cambios que se requiere realizar en las perspectivas de Tecnología, Personas, Procesos?				

14	<b>Objetivos y alcance del proyecto</b>	¿Los objetivos están descritos de manera que cumplan con las características de específico, medible, asignable, realista y a tiempo definido?				
15		¿Se ha descrito al menos un objetivo por cada situación deseada?				
16		¿Está identificado el usuario final del aplicativo?				
17		¿Está especificado a qué Compañía impacta el requerimiento?				
18		¿Están listados todos los productos/servicios, canales de Venta y asociados relacionados a la necesidad?				
19		¿Se describe de qué manera se impacta a cada uno de los productos/servicios, canales de venta y asociados?				
20	<b>Formato</b>	¿El formato del documento está conforme a la última versión de la plantilla de la Solicitud?				
21		¿Se eliminaron las notas que sirven de apoyo para el llenado del documento?				
22		¿Es correcta la ortografía y puntuación del documento?				

Nivel de Calidad del producto	
Puntos máximos	
Puntos Obtenidos	
% de apego	

Tabla 4-9 Plantilla de Lista de Verificación para la Solicitud Inicial (Elaboración Propia)

Una vez que se han registrado los estados de las listas de verificación de Solicitud Inicial, se procede a establecer el nivel de calidad del Producto de acuerdo al puntaje y ponderación del total de los Items. Esto nos dará el porcentaje de cumplimiento del Documento, el cual debe ser 100%.

Los hallazgos identificados deberán estar registrados en el reporte de Hallazgos. El equipo debe realizar el seguimiento respectivo para levantar todas las observaciones.

A continuación en la Tabla 4-11 se muestra la plantilla que deberá ser utilizada para el registro de Hallazgos de la Solicitud Inicial.

[illegible]



#### 4.6.1.2 Planeación: Estimación Inicial

Con la Solicitud Formal Verificada los equipos de desarrollo y calidad podrán realizar las primeras estimaciones de esfuerzo y tiempos involucrados en sus actividades. El equipo de desarrollo elabora el documento de Estimación, Análisis de Impacto y Cronograma; mientras que el Equipo de V&V elabora el documento de Identificación de alcance de V&V y estimación de alto nivel.

La Identificación del alcance de V&V y la estimación de alto nivel se realizan mediante herramientas personalizadas de estimación, Por ejemplo, las basadas en modelos de estimación de proyectos de software como el de Puntos de Función o Puntos de Caso de Uso.

Estos modelos utilizan parámetros de ajustes dependiendo del impacto que tengan en el proyecto los factores técnicos y ambientales de la organización. Los **Pesos** de cada factor son establecidos por el negocio y la complejidad debe ser establecida por la entidad de Verificación y Validación. El nivel de complejidad puede variar entre 1 y 5, siendo 1 el más simple y 5 el más complejo. El **Factor de Complejidad** es el cálculo entre el peso y la complejidad.

En las tablas 4-12 y 4-13 se muestran las plantillas para calificar los factores técnicos y de ambiente.

Factores técnicos			
Factores Técnicos de Proyecto	Peso	Complejidad	Factor de complejidad
Relación y Dependencia con otros Sistemas			
Generación de Data y Reports			
Malla de Procesos / Batch			
Tiempos de Respuesta de la aplicación			
Facilidad de Ambiente de Prueba			
Accesos y Credenciales (Seguridad)			
Se requieren Capacitaciones Funcionales y Técnicas			
Ejecución de Cierres (Diario, Mensual, etc)			

Ejecución de Reportes			
Generación/Validación de Tramas			
<b>Total Factor Técnico</b>			
	<b>Factor Técnico (TCF)</b>		

Tabla 4-11 Estimación Inicial: Factores Técnicos (Elaboración Propia)

<b>3.1 Factores de ambiente</b>			
<b>Factores de Ambiente Organizacional</b>	<b>Peso</b>	<b>Complejidad</b>	<b>Factor de complejidad</b>
Experiencia en la Aplicación			
Documentación del Proceso relacionado			
Facilidad de comunicación entre equipos			
Disponibilidad de Apoyo			
Requerimientos Estables (cerrados)			
<b>Total Factor de Ambiente</b>			
	<b>Factor de Ambiente (EF)</b>		

Tabla 4-12 Estimación Inicial: Factores de Ambiente (Elaboración Propia)

A continuación se debe calificar cada uno de los requerimientos, según el número de impactos y números de flujos asociados. Por cada uno de los requerimientos se registran observaciones que definirán el alcance del proyecto en sí.

En la Tabla 4-14 se muestra la plantilla que se utilizará para la estimación en base a la calificación de requerimientos.

[Solo modificar celdas celestes]		Clasificación Impactos / dependencias				Clasif. Requerimiento			
#	Funcionalidades	# Impactos / Dependencias (Interfases)	Peso	Clasificación	Observaciones	# Flujos / Caminos (Felices / Infelices)	Peso	Clasificación	Observaciones
1									
2									
3									
	Puntos Funcionales sin Ajuste (UUCP)								
	Actividades Adicionales	HORAS							
	Horas necesarias para capacitación								
	Horas necesarias para la Ratificación								

Tabla 4-13 Calificación de Registro de Requerimientos (Elaboración Propia)

Después de completar los factores técnicos, ambientales y calificar los requerimientos se obtiene automáticamente la estimación a alto nivel. En la Tabla 4-15 se muestra la plantilla que se utilizará para la estimación inicial.

#### MODELO ESTIMACIÓN PARA PROYECTOS DE SOFTWARE

**Tipo de Estimación**

**Grado de Dificultad del Proyecto**

#### FACTORES DE ESTIMACION

Puntos Funcionales sin ajuste (UUCP)

Factor de Complejidad Técnica (TCF)

Factor de Complejidad de Ambiente (EF)

**Puntos Funcionales Ajustados (UCP)**

**Esfuerzo Estimado C1 (Horas/Hombre)**

**Ciclo 2**

Gestión y Seguimiento

Capacitaciones

Ratificación

**Desviación**

**Esfuerzo Total**

**ANALISIS**

**DISEÑO**

**CERTIFICACION**

**Horas Lab x Recurso**

**# Recursos**

### **Dias Lineales**

Tabla 4-14 Modelo de Estimación Inicial (Elaboración Propia)

#### 4.6.1.3 Planeación: Plan de V&V

A continuación se muestran los ítems que debe contener el Plan de Verificación y Validación de Software. Se debe tener en cuenta que este plan no es estático, debe cambiar en el tiempo y ser ajustado en el momento en que se tengan los requerimientos funcionales más aterrizados.

En la tabla 4-16 se muestran los puntos que deben ser considerados en el Plan de V&V:

- |  |
|--|
| <ul style="list-style-type: none"><li>• Propósito: Se debe indicar los objetivos del Plan de V&amp;V</li><li>• Contexto</li><li>• Involucrados: Se debe listar los roles, responsables y el servicio o actividad a ejecutar.</li><li>• Aseguramiento y Control de Calidad del Producto: Se debe indicar las principales actividades incluidas en el proceso de Verificación y Validación, especificando los módulos a revisar.</li><li>• Fuera del Alcance: Se debe indicar las exclusiones del proceso.</li><li>• Supuestos: Se debe indicar los supuestos sobre los que se trabajará en el Proceso de Verificación y Validación.</li><li>• Restricciones.</li><li>• Criterios de Aceptación para el Aseguramiento y Control de Calidad del Producto: Indicar cuáles serán los criterios que serán considerados para la aceptación del Software.</li><li>• Plan de Comunicación.</li><li>• Niveles de Escalamiento.</li></ul> |
|--|

Tabla 4-15 puntos a considerar en Plan V&V (Elaboración Propia)

## **4.6.2 Análisis**

### **4.6.2.1 Análisis: Revisión de Requisitos, Especificación Funcional y Técnica**

Para esta Fase de Análisis se requiere que el negocio entregue los requisitos y la Especificación Funcional, cada uno de estos documentos se verificarán utilizando las listas de verificación de Requisitos y Especificación Funcional respectivamente. La lista de verificación sirve para medir el grado de cumplimiento de calidad de este tipo de documentos. Las preguntas listadas en las listas de verificación han sido previamente elaboradas y sirven como guía para realizar la inspección.

En la plantilla se deben registrar los datos del Proyecto y después se debe revisar cada una de la preguntas de las listas de verificación. Para el llenado de esta plantilla se debe considerar las instrucciones. A continuación en la Tabla 4-17 se muestra las listas de verificación que servirá como plantilla para la revisión de Requisitos.

DESCRIPCION	INSTRUCCIONES
ACIERTO	El documento cumple con la verificación
FALLO	El documento no cumple con la verificación
SIN VERIFICAR	No se ha podido verificar si el documento cumple con la verificación
NO APLICA	Para el documento analizado, no aplica la verificación

**PROYECTO:** Control de Inventarios  
**CODIGO PROYECTO:** PRO-0013-0001  
**FASE PROYECTO:** Especificación Funcional  
**DOCUMENTO:** Check\_Requisitos.xls

**FECHA:**  
**TIPO PROYECTO:**  
**REVISADO POR:**  
**VERSION:**

Nº	CUESTION	RESULTADO	OBSERVACIONES
<b>A</b>	<b>Especificación</b>		
A1	Verificar que se realiza una descripción general de la funcionalidad existente u aspectos generales de negocio sobre la base que se realiza la definición de requisitos.		
A2	Verificar que la codificación de la nomenclatura de los requisitos se compone del código del proyecto seguido de un número secuencial, además de una breve título y la descripción que incluya cualquier información necesaria que determine/defina los requisitos y permita su comprensión.		
A3	Verificar que se realiza una lista detallada de los requisitos funcionales o de producto.		
A4	Verificar que los requisitos funcionales se definen de forma clara y concisa con objeto de cubrir un solo aspecto.		
A5	Verificar que se realiza una descripción de la implementación de los procesos batch conforme a los requisitos de explotación/producción de aplicaciones del cliente.		
A6	Verificar que en la descripción de los requisitos de explotación/producción se hace referencia a la nomenclatura del proceso, formato del fichero de log, estructura de directorios, ventana de aplicación, calendario de ejecución, etc.		
A7	Verificar que se tiene en cuenta en la descripción del arranque y parada controlada de un nuevo sistema que éstos deben realizarse conforme los requisitos de explotación de aplicaciones del cliente.		
A8	Verificar que existe una descripción en las nuevas funcionalidades (requisitos) referente a la comprobación del estado de la aplicación y la validación del sistema a nivel de presencia y actividad, así como componentes y funcionalidad.		
A9	Verificar que en algún requisito se indica que, desde el punto de vista de acceso y ejecución de procesos online o batch en el sistema final que soporte la funcionalidad/servicio demandado, el proyecto debe ajustarse a los requisitos de explotación de aplicaciones del cliente		
A10	Verificar que el/los requisito/s referente/es a la gestión de usuarios y contraseñas se realizan conforme a los requisitos de seguridad de aplicaciones del cliente.		



A11	<p>Verificar que en la descripción de los requisitos sobre la gestión de usuarios y contraseñas se incluyen:</p> <ul style="list-style-type: none"> <li>- Identificadores de usuario relacionados con la identidad del usuario (nombre y apellidos, por ejemplo).</li> <li>- Longitud de los identificadores de usuario.</li> <li>- No permitir usuarios genéricos.</li> <li>- Limitar el número de intentos.</li> <li>- Longitud y conjunto de caracteres permitidos para las contraseñas.</li> <li>- Periodo de Validez de la contraseña.</li> <li>- No permitir la repetición de contraseñas cuando caducan.</li> </ul> <p>- Almacenamiento y transmisión encriptada de contraseñas.</p>		
A12	Verificar que se incluye un requisito referente a las copias de seguridad, indicando su periodicidad y recuperación.		
A13	Verificar que se incluye un requisito referente al borrado de ficheros temporales, indicando su periodicidad.		
A14	Verificar que se incluye un requisito referente a las pruebas con datos reales.		
A15	Verificar que se incluye un requisito sobre el código fuente del desarrollo, describiendo la entrega del código fuente kernel y no kernel cuando se haya acordado.		
A16	Verificar que se realiza un requisito sobre Ejecutables, indicando la entrega de ejecutables, binarios, librerías, etc., tanto kernel como no kernel necesarios para el correcto funcionamiento.		
A17	Verificar que se realiza un requisito sobre ficheros auxiliares, indicando la entrega de ficheros auxiliares de configuración, scripts, etc., necesarios para el correcto funcionamiento.		
A18	Verificar que se realiza un requisito sobre el entorno servidor, indicando en su descripción de la obligación del correcto funcionamiento de los desarrollos en entorno servidor y con los sistemas operativos del cliente.		
A19	Verificar que se realiza un requisito sobre el entorno cliente, indicando en su descripción de la obligación del correcto funcionamiento de los desarrollos sobre las máquinas y con los sistemas operativos del cliente.		
A20	Verificar que se realiza un requisito sobre el procedimiento de instalación en los distintos entornos en los que se instale la solución.		
A21	Verificar que se realiza un requisito referente a la responsabilidad del despliegue de la aplicación, tanto a nivel de servidor como de cliente.		
A22	Verificar que se realiza un requisito correspondiente al tiempo requerido de instalación, indicando también todo aquello que implique indisponibilidad del sistema o impacto en otras funcionalidades.		
A23	Verificar que se realiza un requisito acerca de los entregables de documentación mínimos al cliente.		
A24	Verificar que se realiza un requisito acerca de los posibles entregables de documentación adicionales.		
A25	Verificar que se realiza un requisito indicando que el análisis y desarrollo del proyecto solicitado debe ser funcionalmente completo, incluyendo las modificaciones necesarias sobre todas las funcionalidades existentes en el sistema o en curso.		

A26	Verificar que se realiza un requisito indicando que todas las cadenas y procesos existentes o en curso, deben seguir funcionando correctamente tras la implantación de la solución propuesta.		
A27	Verificar que se realiza un requisito que indique la mejora de prestaciones del sistema.		
A28	Verificar que se realiza un requisito en el que se contemple la posibilidad de ejecutar pruebas de volumen para la funcionalidad y/o proceso, siempre que sea posible.		
A29	Verificar que se realiza un requisito en el que se plantee una opción con la solución kernel para implementar la funcionalidad solicitada, siempre que sea posible.		
A30	Verificar que se realiza un requisito en el que se indique la intención del proyecto de reducir las necesidades del sistema.		
A31	Verificar que se realiza un requisito en el que se contemple la posibilidad de ejecutar pruebas de volumen para la funcionalidad y/o proceso, siempre que sea posible.		
A32	Verificar que se realiza un requisito en el que se describa la limpieza de objetos obsoletos.		
A33	Verificar que se realiza un requisito en el que se describa la gestión completa de errores, quedando siempre reflejada en el EV la forma de tratar los posibles errores generados.		
A34	Verificar que se realiza un requisito indicando que el proyecto debe funcionar correctamente sobre el entorno de producción, incluyendo la corrección o recuperación ante inconsistencias de datos, configuración del entorno o volumen real de datos		
A35	Verificar que se realiza un requisito en el que se contemple la posibilidad de ejecutar pruebas de volumen para la funcionalidad y/o proceso, siempre que sea posible.		
A36	Verificar que se realiza un requisito que indique la independencia de máquina y alta disponibilidad, con el software de máquina preparado para fail over automático entre máquinas y sin limitaciones funcionales que obliguen a intervenciones manuales.		
A34	Verificar que se realiza un requisito que indique la/as alternativa/as que tenga menor impacto en la disponibilidad del sistema, al hora de la puesta en producción del proyecto.		
A34	Verificar que realiza un requisito que indique la obligatoriedad de no incluir ninguna acción manual de soporte en el proyecto.		
<b>B</b>	<b>Otros</b>		
B1	El documento de Requisitos se ha elaborado en la fase correspondiente del proyecto.		
B2	Los documentos asociados a los requisitos se mantienen actualizados a lo largo del proyecto.		
B3	Se dispone, tras la toma de requisitos, de evidencia de la Revisión de los mismos (RQ actualizado y aprobado en Newark).		

Tabla 4-16 Listas de verificación de Requisitos (Elaboración Propia)

Si después de llenar la plantilla de las listas de verificación de Requisitos, se encuentran uno o más registros con estado: Fallo, entonces el negocio debe levantar las observaciones reportadas. Este ciclo es repetitivo y finaliza cuando las observaciones han sido levantadas al 100% o cuando el negocio decide terminarlo por alguna política.

A continuación en la Tabla 4-18 se muestra las listas de verificación que servirá como plantilla para la revisión de la Especificación Funcional.

DESCRIPCION	INSTRUCCIONES
ACIERTO	El documento cumple con la verificación
FALLO	El documento no cumple con la verificación
SIN VERIFICAR	No se ha podido verificar si el documento cumple con la verificación
NO APLICA	Para el documento analizado, no aplica la verificación

**PROYECTO:**

**CODIGO PROYECTO:**

**FASE PROYECTO:**

**DOCUMENTO:**

**FECHA:**

**TIPO PROYECTO:**

**REVISADO POR:**

**VERSION:**

Nº	CUESTION	RESULTADO	OBSERVACIONES
<b>A</b>	<b>Introducción</b>		
A1	Verificar que se incluye un resumen del objetivo de la aplicación.		
A2	Verificar que se incluye un resumen de la funcionalidad de la aplicación.		
<b>B</b>	<b>Análisis Funcional</b>		
B1	Verificar que se listan las funcionalidades que tendrá que cubrir la aplicación.		
B2	Verificar la existencia de la descripción de flujos de negocio.		
B3	Verificar la existencia de la descripción de módulos funcionales.		
B4	Verificar la existencia de la descripción de flujos de pantallas.		
<b>C</b>	<b>Configuración</b>		
C1	Verificar que se realiza una descripción de parámetros configurables a nivel de negocio.		
C2	Verificar que se realiza una descripción de parámetros configurables a nivel de aplicación.		
C3	Verificar que se realiza una descripción de perfiles de		

	usuario.		
<b>D</b>	<b>Performance</b>		
D1	Verificar que se realiza una definición de los niveles mínimos de performance de los procesos.		
D2	Verificar que se incluye una estimación del volumen de usuarios y datos.		
D3	Verificar que se incluyen posibles concurrencias.		
<b>E</b>	<b>Política de seguridad</b>		
E1	Verificar que se realiza una definición de niveles mínimos respecto a la restricción del acceso a la aplicación.		
E2	Verificar que se realiza una definición de niveles mínimos respecto al acceso a los datos.		
E3	Verificar que se realiza una definición de política de backup y copias de seguridad.		
E4	Verificar que se incluye una política de accesos.		
<b>F</b>	<b>Operación y mantenimiento</b>		
F1	Verificar que se realiza una descripción de la administración y configuración de la aplicación.		
F2	Verificar que se realiza una descripción de procesos		

	externos y monitorización (descripción, periodicidad...).		
F3	Verificar que se realiza una descripción de la gestión de usuarios.		
F4	Verificar que se incluye una descripción de la generación de logs y estadísticas.		
<b>G</b>	<b>Otros</b>		
E12	El documento de Análisis Funcional se ha elaborado en la fase correspondiente del proyecto.		
E11	Los documentos asociados al diseño se mantienen actualizados a lo largo del proyecto.		
E12	Se dispone, tras la fase de diseño, de evidencia de la Revisión del Diseño (AF/AT actualizados y aprobados en Newark).		
<b>PROPUESTAS MEJORA</b>			
1			
2			
3			
<b>CONCLUSIONES</b>			

Tabla 4-17 Listas de verificación de Especificación Funcional (Elaboración Propia)

Si después de llenar la plantilla de listas de verificación de Especificación Funcional, se encuentran uno o más registros con estado: Fallo, entonces el negocio debe levantar las observaciones reportadas. Este ciclo es repetitivo y finaliza cuando las observaciones han sido levantadas al 100% o cuando el negocio decide terminarlo por alguna política.

Cuando la Especificación Funcional ha sido revisada, el área de desarrollo inicia con la construcción de la Especificación Técnica. Cuando la elaboración de este documento finaliza es entregado al área de Calidad para que pase por el proceso de Verificación respectivo.

A continuación en la Tabla 4-19 se muestra las listas de verificación que se utilizará para la revisión de este tipo de documento.

DESCRIPCION	INSTRUCCIONES
ACIERTO	El documento cumple con la verificación
FALLO	El documento no cumple con la verificación
SIN VERIFICAR	No se ha podido verificar si el documento cumple con la verificación
NO APLICA	Para el documento analizado, no aplica la verificación

**PROYECTO:**

**CODIGO PROYECTO:**

**FASE PROYECTO:**

**DOCUMENTO:**

**FECHA:**

**TIPO PROYECTO:**

**REVISADO POR:**

**VERSION:**

Nº	CUESTION	RESULTADO	OBSERVACIONES
<b>A</b>	<b>Introducción</b>		
A1	Verificar que se incluye un resumen del objetivo de la aplicación.		
A2	Verificar que se incluye un resumen de la funcionalidad de la aplicación.		
<b>B</b>	<b>Arquitectura</b>		
B1	Verificar que se incluye un diagrama de componentes HW y SW.		
B2	Verificar que se realiza una descripción de la arquitectura del sistema.		
B3	Verificar que se realiza una descripción de todos los componentes que constituyen la arquitectura.		
B4	Verificar que se realiza una descripción de los entornos implicados.		
<b>C</b>	<b>Componentes</b>		
C1	Verificar que se realiza una descripción completa de todos los componentes que conforman la aplicación.		
<b>D</b>	<b>Procesos</b>		
D1	Verificar que se realiza una descripción completa de los procesos, batch, tiempo real, etc que constituyen el funcionamiento de la aplicación.		
<b>E</b>	<b>Integración con otros sistemas</b>		
E1	Verificar que se realiza una descripción de los flujos de entrada y de salida de datos.		
E2	Verificar que se realiza una descripción de los formatos de los Inputs y Outputs del sistema provenientes o dirigidos hacia otros sistemas del mapa de sistemas del cliente.		
E3	Verificar que se realiza una descripción de procesos vinculados.		
<b>F</b>	<b>Otros</b>		
E12	El documento de Análisis Técnico se ha elaborado en la fase correspondiente del proyecto.		
E11	Los documentos asociados al diseño se mantienen actualizados a lo largo del proyecto.		
E12	Se dispone, tras la fase de diseño, de evidencia de la Revisión del Diseño (AF/AT actualizados y aprobados en Newark).		
<b>PROPUESTAS MEJORA</b>			



1	
CONCLUSIONES	

Tabla 4-18 Lista de Verificación de Especificación Técnica (Elaboración Propia)

### **4.6.3 Diseño**

#### **4.6.3.1 Casos de Pruebas**

Es importante que el caso de pruebas este correctamente identificado, así también se deben colocar los pasos indican como ejecutar los casos de prueba.

Los campos utilizados fueron el código del caso, la aplicación, modulo o componente de prueba, la descripción del caso de prueba, el grado de complejidad para determinar el tiempo que podría tardar la ejecución, las pre-condiciones que deben cumplirse para ejecutar el caso de prueba, los pasos detallados para poder ejecutar el caso y los puntos de verificación o resultados esperados para cada paso.

Se debe colocar el resultado esperado de cada paso y también de todo el caso de prueba. El Tester puede colocar algún comentario que permita ampliar el detalle del caso de prueba. A continuación en la tabla 4-20 se muestra la plantilla utilizada para el registro de los casos de pruebas.

ID y Nombre del Requerimiento	[Número y Nombre del Requerimiento]
Líder del Proyecto	[Nombre del Líder del Proyecto]
Administrador de Calidad	[Nombre del Administrador de Calidad]
Líder Técnico de Calidad	[Nombre de la persona que generó el documento]

Versión del REF con la que se genera Diseño	[Indicar versión del documento]
Versión del RET con la que se genera Diseño	[Indicar versión del documento]

**Objetivo:**

[Indicar el objetivo de las pruebas a realizar]

ID	Aplicación / Módulo / Componente	Nombre del caso de Prueba (Flujo Básico / Alternativo)	Para Ratificación	Complejidad	Precondicio nes	Postcondicio nes	Pasos	Puntos de Verificación (PV) / Resultado Esperado	Ciclo 1 de Pruebas de Certificación		
									Periodo de ejecución: [dd/mm/aaaa - dd/mm/aaaa]		
									Ingeniero de Pruebas: [Nombre de la persona que hizo la verificación]		
									Resultado del PV	Resultado del Guión de Prueba	ID Defecto / Comentarios
1							1.	1			
							2.	2			
							3.	3			
							4.	4			
							5.	5			
2							1.	1			
							2.	2			
							3.	3			
							4.	4			
							5.	5			

Tabla 4-19 Plantilla de Casos de Prueba (Elaboración Propia)

#### 4.6.4 Pruebas y Ratificación

##### 4.6.4.1 Ejecución de Pruebas

Después de que el desarrollo del software y las pruebas unitarias concluyen, se inicia la ejecución de las pruebas. Este punto se realiza después de haber revisado el correcto funcionamiento del ambiente de pruebas y haber obtenido la data con la que se realizarán las pruebas. Para esta etapa los casos de prueba son ejecutados paso a paso, el campo **Resultado del Guion de Prueba** se deja vacío cuando el caso de prueba aún no es ejecutado, otros valores del estado dependerán del resultado de la ejecución:

- **Bloqueado:** Cuando la ejecución del caso de prueba depende de otros casos de prueba que no se pueden ejecutar o alguna pre-condición no se cumple.
- **Paso:** Cuando la ejecución del caso de prueba se ha realizado correctamente en todos sus pasos.
- **Fallo:** Cuando la ejecución del caso de prueba ha fallado en alguno de sus pasos.

Cuando un caso de pruebas o varios casos de prueba han fallado, también se debe llenar la plantilla de **Reporte de Defectos** o incidencias. Este reporte permite llevar el control y realizar seguimiento a la solución de los defectos. El registro de defectos incluye: un código, el número de ciclo de pruebas, la aplicación y módulo de pruebas, el código del caso de pruebas, descripción del defecto, la fecha en la que se detectó el defecto, la severidad (alta, medio, bajo), la prioridad de solución que debe ser revisado en conjunto con el negocio, el estado (abierto, solucionado, cerrado) y el tipo de defecto (si es de forma o funcional).

Las otras columnas sirven para registrar las acciones correctivas y para llevar un control y seguimiento sobre la solución del defecto.

En la Tabla 4-21 se muestra la plantilla de Reporte de Defectos que sirve para registrar todos los defectos/incidencias encontrados durante las pruebas.



En el presente Capítulo IV se realizó una descripción del Marco Metodológico, en el siguiente Capítulo V se muestra el caso de estudio, es decir la aplicación del Marco Metodológico propuesto en una empresa Pyme.

## Capítulo 5: Caso de estudio

En el presente capítulo se muestra el caso de estudio, que es una Pyme Peruana en donde se aplicó el Marco Metodológico propuesto para el Proceso de Verificación y Validación de Software. Primero se describen las características de la empresa, se muestra el esfuerzo de instalación requerido para aplicar el Marco Metodológico, se aplicó cada uno de los procesos de Verificación y Validación al software elegido (descritos en el Capítulo 4, incluyendo los formatos y plantillas que fueron utilizados) y finalmente se muestran los resultados obtenidos.

### 5.1 Características de la empresa

Nuestro caso de estudio tomará como referencia a la Empresa Q-BOSS, una PYME que inicia operaciones a mediados del 2011 y brinda soluciones informáticas. Actualmente la empresa cuenta con 9 empleados, de los cuales seis de ellos se dedican a la parte de desarrollo, operación y mantenimiento de los productos de software.

La gerencia ha decidido apostar por la mejora de procesos para dar soporte a la organización, consolidación y crecimiento de la empresa de manera sistemática. La empresa no tiene experiencia en la mejora de procesos software y no cuenta con profesionales capacitados en lo que a calidad de software se refiere, sin embargo existe el deseo de mejorar sus procesos para ofrecer a sus clientes productos y servicios de mayor nivel de calidad. Es por ello, que se ha comenzado un ciclo de mejora de procesos con el soporte de una asesora.

La empresa seleccionó el **Proyecto de Control de Inventarios** para aplicar el Marco Metodológico propuesto.

### 5.2 Instalación y esfuerzo del Marco Metodológico propuesto

Para la instalación del ciclo de mejora se recopiló toda la información de la empresa y se firmó una carta de compromiso. En la reunión inicial la empresa reconoció que su proceso para desarrollar software es caótico. El objetivo es institucionalizar el Marco Metodológico propuesto. Se ha definido el cronograma

general para la aplicación marco metodológico teniendo en cuenta que el tiempo total es de 20 semanas.

En la reunión de lanzamiento la asesora compartió qué trabajo se va a realizar en la empresa y se retroalimentó con las expectativas que tienen los empleados con el proyecto. El trabajo se basó en los siguientes puntos: Análisis del estado inicial de la empresa, revisión de sus procesos actuales y aplicación del Marco Metodológico propuesto al Sistema elegido.

El esfuerzo involucrado en la realización de esta actividad se presenta en la siguiente tabla 5-1:

Consultora	Empresa
1 persona por 245 minutos	1 persona por 240 minutos

Tabla 5-1 Esfuerzo realizado para la actividad de instalación (Elaboración Propia)

### **5.3 Análisis del estado inicial de la empresa.**

La empresa cuenta con personal de Sistemas, pero dependiendo de la complejidad de los requerimientos o la disponibilidad de su personal puede optar por sub contratar servicios de desarrollo tanto para brindar mantenimiento a los sistemas existentes como para implementar nuevas funcionalidades o nuevos sistemas. La empresa se visitó por un periodo de 2 días para realizar la valoración inicial de los procesos escogidos. Para valorar estos procesos se aplicaron técnicas de recolección de evidencias entrevista y encuestas. Debido a que la empresa no tenía documentación sobre sus procesos, se realizó una reunión con la persona responsable de los procesos para levantar el diagrama de actividades de estos procesos. A continuación en la tabla 5-2 se muestra el esfuerzo realizado para la actividad de análisis:



Consultora	Empresa
1 persona por 370 minutos	1 persona por 1140 minutos

Tabla 5-2 Esfuerzo realizado para la actividad de análisis (Elaboración Propia)

Como se puede apreciar en la tabla 5-3 la empresa no aplica ningún proceso de Verificación y Validación estática o dinámica

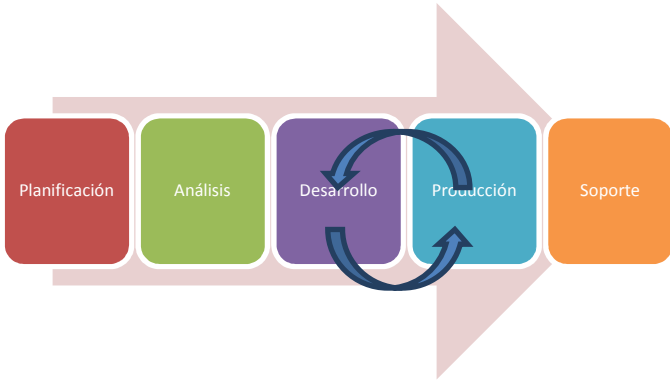
Proceso Actual	Características
	<ul style="list-style-type: none"> <li>• Tienen cinco fases: planificación, análisis, desarrollo, puesta en producción y soporte.</li> <li>• Sólo se realizan pruebas unitarias pero ninguna actividad de V&amp;V de software. Estas pruebas unitarias son ejecutadas por los programadores.</li> <li>• Se evidenció que en el 89% de los aplicativos que entran a producción reportan costos de mantenimiento altos.</li> <li>• Existe un 85% de llamadas a Help Desk ocasionadas por fallas en el Software.</li> <li>• Clientes insatisfechos por constantes cortes de servicio.</li> </ul>

Tabla 5-3 Proceso Actual de Desarrollo – Pruebas de QBOSS (Elaboración Propia)

## 5.4 Aplicación del Marco Metodológico del Proceso de Verificación y Validación.

El primer paso para la aplicación del marco metodológico es establecer el alineamiento entre las fases y las entidades involucrados, así como establecer responsabilidades y entregables.

La tabla 5-4 muestra el esfuerzo dedicado a las actividades de formulación y ejecución de mejoras.

Consultora	Empresa
1 persona por 1500 minutos	1 persona por 3000 minutos

Tabla 5-4 Esfuerzo realizado para la actividad de aplicación del Marco Propuesto (Elaboración Propia)

La empresa aplicó el marco metodológico propuesto para cada una de los procesos seleccionados.

### 5.4.1 Planeación

Esta primera fase se inició con la solicitud formal del negocio, para el caso de estudio se realizó una verificación de la Solicitud de Implementación del Software de Control de Inventarios. Esta necesidad fue verificada por el equipo de calidad para asegurar que el documento se encontraba alineado a lo que realmente necesitaba el usuario.

#### 5.4.1.1 Documento de solicitud formal

A continuación se muestra la solicitud inicial del Proyecto (ver tabla 5-5):

#### SOLICITUD - SECCIÓN INICIAL

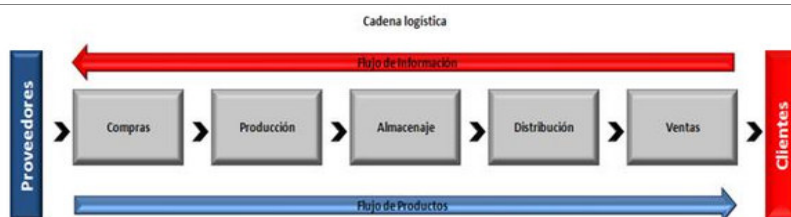
##### DATOS GENERALES

Nombre del Proyecto:	Nuevo Sistema de Control de Inventarios
Unidad del Solicitante:	Logística
Usuario Sponsor:	Luis Chavarri
Líder Usuario:	Maria Fernanda Valenzuela
Beneficiarios:	Logística - Operaciones
Especialista del Negocio:	Susana Ancelmo
Fecha de Creación:	28-nov

Tipo de Requerimiento:	Nuevo Desarrollo
Fecha de Aprobación en el Comité de Proyectos	02 – Dic

## PROCESO

### Diagrama del Proceso Actual



Proceso Relacionados directamente	#	Macroprocesos	Procesos	Sub - Proceso (Actividad)	Código de Documento <sup>(1)</sup>
	1	Proceso Logístico	Abastecimiento	Compra/ Reposición	
	1	Proceso Logístico	Almacenamiento	Organización de anaqueles	
Proceso Relacionados indirectamente	Nº	Sub - Proceso (Actividad)	Descripción del Impacto		
	1	Venta	El Modulo de venta deberá comunicarse con el nuevo sistema de control de inventarios		
	2				
	3				

## DESCRIPCIÓN DE LA NECESIDAD

Diagnóstico de la Situación Actual	1.-	El control de inventarios se realiza actualmente en forma manual lo que genera perdidas
	2.-	Se ha incrementado las mermas y las entregas de productos a los clientes se ven retrasadas o simplemente no se cumplen causando una total insatisfacción al cliente

Resumen del Análisis Realizado	1	Se requiere un nuevo sistema que logre automatizar el proceso y que brinde información certera en línea para tomar decisiones y para que otros sistemas consulten los stocks actuales.			
	2				
	3				

#### ALINEAMIENTO ESTRATÉGICO

Forma Parte de un Proyecto Estratégico de la Compañía	SI/NO	Código y Nombre del Proyecto Estratégico
	NO	
Forma Parte de un Proyecto Estratégico de la División	SI	Re-estructuración del Área de Logística
Impacto / Alineamiento a los Pilares de Gestión		
Pilar de Gestión	SI/NO	Justificación del alineamiento
Enfoque en el Cliente	SI	Asegurar la satisfacción del cliente con la entrega oportuna de sus productos
Canales Altamente Productivos y Rentables		
Servicio Extraordinario y Eficiente		
Solidez Financiera		

Tabla 5-5 Formato de Solicitud Inicial (Elaboración Propia)

Esta solicitud de atención pasó por un proceso de verificación, con la finalidad de asegurar el cumplimiento de estándares requeridos, así como asegurar que lo solicitado por el usuario era coherente, congruente y alineado a sus necesidades.

En la Tabla 5-6 se muestra las listas de verificación de Inspección utilizado. El porcentaje de apego fue de 79%, por ello se registraron los hallazgos encontrados en esta fase.

No.	Sección	Punto a Verificar	Aplica Criterio	Ponderación	Puntuación	Estado
1	Datos Generales	¿Se ha registrado el Nombre del Proyecto, Unidad del Solicitante, Usuario Sponsor, Líder Usuario e Involucrados?	Si	1	1	1
2		¿Están listadas el Área, rol, nombre, las responsabilidades de los involucrados CI?	Si	2	4	2
3		¿Está definido el Nombre del Especialista de Negocio responsable del SOL inicial?	Si	1	3	3
4		¿Se ha incluido la fecha de creación de la solicitud Inicial?	Si	1	3	3
5		¿Se ha incluido la fecha en la que se aprobó en el Comité de Gobierno de TI?	No	2	0	
6	Proceso	¿El diagrama de proceso actual permite visualizar el flujo actual de trabajo?, si es una link, ¿esta funciona correctamente.	Si	3	3	1
7		¿El diagrama de proceso indica el flujo, los actores y los productos de manera clara?	Si	3	3	1
8		En caso de que el proceso no sea afectado, ¿se está puntualizando que no aplica, o que no se cuenta con el diagrama del proceso?	No	2	0	
9	Análisis de Brechas	La descripción de la situación actual, ¿Permite identificar las causas de la necesidad del negocio en las perspectivas de Tecnología, Personas, Procesos?	Si	3	9	3
10		¿La situación actual está descrita de manera concisa? (Esta especificada de forma clara y sencilla; y solamente se incluye un problema u oportunidad en cada fila)	Si	3	6	2
11		¿La descripción de la situación deseada es clara y sin ambigüedades, y describe lo que se desea obtener en las perspectivas de Tecnología, Personas, Procesos?	Si	3	6	2
12		¿La descripción de la situación deseada es completa? (Es decir contiene un resultado deseado por cada fila, no son redundantes y especifica la totalidad del resultado deseado).	Si	3	6	2
13		Al comparar la situación actual con la situación deseada, ¿Se pueden identificar los cambios que se requiere realizar en las perspectivas de Tecnología, Personas, Procesos?	Si	3	9	3

14	Objetivos y alcance del proyecto	¿Los objetivos están descritos de manera que cumplan con las características de específico, medible, asignable, realista y a tiempo definido?	Si	3	9	3
15		¿Se ha descrito al menos un objetivo por cada situación deseada?	Si	3	9	3
16		¿Está identificado el usuario final del aplicativo?	Si	1	3	3
17		¿Está especificado a qué Compañía impacta el requerimiento?	Si	1	3	3
18		¿Están listados todos los productos/servicios, canales de Venta y asociados relacionados a la necesidad?	Si	3	9	3
19		¿Se describe de qué manera se impacta a cada uno de los productos/servicios, canales de venta y asociados?	Si	1	3	3
20	Formato	¿El formato del documento está conforme a la última versión de la plantilla de la Solicitud?	Si	3	9	3
21		¿Se eliminaron las notas que sirven de apoyo para el llenado del documento?	Si	1	2	2
22		¿Es correcta la ortografía y puntuación del documento?	Si	1	2	2

Nivel de Calidad del producto	
Puntos máximos	141
Puntos Obtenidos	112
% de apego	79%

Tabla 5-6. Listas de verificación de Solicitud Inicial (Elaboración Propia)

En la tabla 5-7 se muestra el historial de hallazgos encontrados.

Registro de Hallazgos								
A ser llenado por el Revisor (es) asignado (s)								
Id. Hallazgo	Etapas	Punto de Verificación	Descripción de Hallazgo	Fecha Identificación	Severidad	Impacto	Tipo	Estado
1	Inicial	2	No se encuentra el ROL especificado de los involucrados en la Sección de "Datos Generales".	03/12/2014	Media	Medio	Falta de Información	Abierto
2	Inicial	7	No se encuentra el Proceso Actual definido.	03/12/2014	Alto	Alto	Falta de Información	Abierto
3	Inicial	8	No se encuentra definido el nuevo Proceso.	03/12/2014	Alto	Alto	Falta de Información	Abierto
4	Inicial	11	El Punto 3 de la Situación Actual está incluido ya en el Punto 2, esto para un mejor entendimiento.	03/12/2014	Media	Medio	Falta de Información	Abierto
5	Inicial	12, 13	Punto 2 de la Situación Deseada debería ir en dos filas para tener un mayor entendimiento.	03/12/2014	Media	Medio	Falta de Información	Abierto
6	Inicial	25	La sección HISTORIAL no se llenó.	03/12/2014	Baja	Bajo	Falta de Información	Abierto
7	Inicial	26	Hay Notas de Ayuda que no han sido borradas	03/12/2014	Baja	Bajo	Forma	Abierto
8	Inicial	27	Hay palabras escritas incorrectamente.	03/12/2014	Baja	Bajo	Forma	Abierto



9	Inicial	-	El campo "Nombre de Revisor" está Vacío.	03/12/2014	Baja	Bajo	Falta de Información	Abierto
10	Inicial	-	No se llenaron los campos Vacíos según Corresponda: con el Texto NO APLICA	03/12/2014	Baja	Bajo	Forma	Abierto

Tabla 5-7 Registro de Hallazgos (Elaboración Propia)

Con la Solicitud Formal Verificada los equipos de desarrollo y calidad realizaron las primeras estimaciones de esfuerzo y tiempos. A continuación en la tabla 5-8 se muestra la calificación de los factores técnicos:

<b>Factores técnicos</b>			
<b>Factores Técnicos de Proyecto</b>	<b>Peso</b>	<b>Complejidad</b>	<b>Factor de complejidad</b>
Relación y Dependencia con otros Sistemas	2	1	2
Generación de Data y Reports	1	2	2
Malla de Procesos / Batch	1	0	0
Tiempos de Respuesta de la aplicación	0,5	1	0,5
Facilidad de Ambiente de Prueba	2	1	2
Accesos y Credenciales (Seguridad)	1	1	1
Se requieren Capacitaciones Funcionales y Técnicas	1	1	1
Ejecución de Cierres (Diario, Mensual, Siniestros, etc)	2	0	0
Ejecución de Reportes	1	3	3
Generación/Validación de Tramas	1	4	4
<b>Total Factor Técnico</b>			<b>15,5</b>
	<b>Factor Técnico (TCF)</b>		<b>0,76</b>

Tabla 5-8 Factores Técnicos (Elaboración Propia)

A continuación en la tabla 5-9 se muestra la calificación de los factores de ambiente:

<b>3.1 Factores de ambiente</b>			
<b>Factores de Ambiente Organizacional</b>	<b>Peso</b>	<b>Complejidad</b>	<b>Factor de complejidad</b>
Experiencia en la Aplicación	2	4	8
Documentación del Proceso relacionado	1	4	4
Facilidad de comunicación entre equipos	0,5	4	2
Disponibilidad de Apoyo	2	3	6
Requerimientos Estables (cerrados)	1	5	5
<b>Total Factor de Ambiente</b>			<b>25,00</b>
	<b>Factor de Ambiente (EF)</b>		<b>0,65</b>

Tabla 5-9 Factores de Ambiente (Elaboración Propia)

A continuación en la tabla 5-10 se calificó cada uno de los requerimientos, según el número de impactos y números de flujo asociados.

[Solo modificar celdas celestes]		Clasificación Impactos / dependencias				Clasif. Requerimiento			
#	Funcionalidades	# Impactos / Dependencias (Interfases)	Peso	Clasificación	Observaciones	# Flujos / Caminos (Felices / Infelices)	Peso	Clasificación	Observaciones
1	Módulo de Planeamiento de Requerimientos de Materiales. - Pedidos Automáticos. - Control de Perecibles InventSys & SaleMax	2	2	MEDIO		4	10	MEDIO	
2	Almacenaje - Organización de Anaqueles - Control RFID InventSys & SaleMax	2	2	MEDIO		2	5	SIMPLE	
3	Generación de Reportes de Control - Cantidad de mermas x mes - Productos más vendidos	2	2	MEDIO		8	15	COMPLEJO	

- Cantidad de solicitudes no cumplidas InventSys & SaleMax								
	8				40			
Puntos Funcionales sin Ajuste (UUCP)	48							
Actividades Adicionales	HORAS							
Horas necesarias para capacitación	0							
Horas necesarias para la Ratificación	5							

Tabla 5-10 Registro de Requerimientos (Elaboración Propia)

Después de completar los factores técnicos (ver Tabla 5-8), ambientales (ver Tabla 5-9) y calificar los requerimientos (ver Tabla 5-10), se obtiene automáticamente lo siguiente (ver Tabla 5-11):

<b>MODELO ESTIMACIÓN UCP PARA PROYECTOS DE DESARROLLO</b>	
<b>Sistema de Control de Inventarios</b>	
<b>Tipo de Estimación</b>	<b>REGULAR</b>
<b>Grado de Dificultad del Proy</b>	<b>MEDIO</b>
<b>FACTORES DE ESTIMACION</b>	
Puntos Funcionales sin ajuste (UUCP)	48,00
Factor de Complejidad Técnica (TCF)	0,76
Factor de Complejidad de Ambiente (EF)	0,65
<b>Puntos Funcionales Ajustados (UCP)</b>	<b>23,56</b>
<b>Esfuerzo Estimado C1 (Horas/Hombre)</b>	<b>235,56</b>
<b>Ciclo 2</b>	70% 90,69
Gestión y Seguimiento	60,00 60,00
Capacitaciones	0,00
Ratificación	5,00
<b>Desviación</b>	<b>20%</b>
<b>Esfuerzo Total</b>	<b>469,50</b>

<b>ANALISIS</b>	15,00%	35,33
<b>DISEÑO</b>	30,00%	70,67
<b>CERTIFICACION</b>	55,00%	129,56
	<b>100,00%</b>	<b>235,56</b>

<b>Horas Lab x Recurso</b>	<b>8</b>
<b># Recursos</b>	<b>2</b>

Tabla 5-11 Modelo de Estimación (Elaboración Propia)

Cuando la solicitud Formal llegó al 100% de Apego y se obtuvo el alcance y estimación inicial (alto nivel), se inició con la elaboración del Plan de V&V. El Plan de V&V para el proyecto es el mostrado a continuación.

#### 5.4.1.2 Plan de V&V

A continuación se muestra el plan de V&V utilizado en el Proyecto

##### 1. Propósito

Para el proyecto Nuevo Sistema de “Control de Inventarios”, el presente documento soporta los siguientes objetivos:

- Identificar la información existente en el Proyecto.
- Listar las funcionalidades a probar según el Reporte de Especificaciones Funcionales (REF).
- Definir la estrategia de pruebas que debe emplearse.
- Identificar los recursos necesarios para ejecutar las actividades planeadas.
- Listar los artefactos entregables del proceso de Aseguramiento y Control de Calidad del Producto.

##### 2. Contexto

Se define este Plan de V&V que corresponde con el Proyecto de Tecnología “Nuevo Sistema de Control de Inventarios”, consistente en la creación de un nuevo sistema web que permita la gestión de los inventarios en forma automatizada.

##### 3. Involucrados

A continuación en la tabla 5-12 se muestra la lista de involucrados, los roles, los responsables

<b>Rol</b>	<b>Responsable</b>	<b>Servicio/Empresa</b>
Líder de Proyecto	<b>Antonio Valcarcel</b>	<b>QBoss</b>
Líder Usuario	<b>Maria Fernanda Valenzuela</b>	<b>Empresa Cliente</b>
Especialista de Negocio	<b>Susana Ancelmo</b>	<b>Empresa Cliente</b>

Tabla 5-12 Involucrados del Proyecto (Elaboración Propia)

#### 4. Aseguramiento y Control de Calidad del Producto

Para este requerimiento son parte del alcance los siguientes puntos:

- Revisiones Formales
  - Revisión formal de la Solicitud de atención
  - Revisión formal de las Especificaciones Funcionales (REF).
  - Revisión formal de las Especificaciones Técnicas (RET).

- Fuera del Alcance

Para este requerimiento, quedan fuera del alcance los siguientes puntos:

- Cualquier otro documento que no se encuentre listado en el punto anterior, como: manuales de operación, instalación y de Usuario.
- Alcance de las actividades de Control de Calidad del Producto

Para este requerimiento son parte del alcance la ejecución de pruebas funcionales para verificar la correcta implementación de las siguientes funcionalidades:

- Módulo de Planeamiento de Requerimientos de Materiales
  - Verificar que se realicen Pedidos Automáticos.
  - Verificar que se lleve a cabo el Control de Perecibles.
- Almacenaje

- Verificar que esté funcionando el módulo de Organización de anaqueles
- Verificar que se esté realizando el control RFID
- Generación de Reportes de Control
  - Verificar Cantidad de mermas x mes Módulo de Venta
  - Verificar Productos más vendidos
  - Cantidad de solicitudes no cumplidas

## 5. Fuera del Alcance

Para este Proyecto, queda fuera del alcance los siguientes puntos:

- Requerimientos no detectados desde la fase del análisis y que a raíz de esto, no se hayan identificado todos los posibles escenarios y por consecuencia no se hayan implementado en la solución.
- Cualquier modificación en el sistema que este fuera del alcance de la Solicitud inicial.
- Configuraciones que se encuentren fuera del alcance del proyecto.
- Validación a reportes, log de errores, o excepciones adicionales que no estén incluidas en la Solicitud inicial.
- Cualquier punto no especificado en este documento.
- Pruebas o defectos relacionados a los errores conocidos.

## 6. Supuestos

- El proceso de Verificación y Validación, no puede ser omitido bajo ninguna circunstancia sin autorización del Gerente de Proyectos.
- Que se haya proporcionado las últimas versiones aprobadas de la Solicitud de atención, de las Especificaciones Funcionales y las Especificaciones Técnicas.
- Los tiempos para las actividades de control de calidad serán los establecidos en el cronograma de trabajo, dichos tiempos deberán ser respetados independientemente de que se presente algún



retraso en la entrega de documentación y/o productos indispensables para la ejecución de las mismas.

- Se debe de contar con un ambiente homologado de pruebas exclusivamente de certificación para realizar las pruebas funcionales a todos los sistemas afectados. Dicho ambiente debe ser replicado en producción o desarrollo en el caso que corresponda.

## 7. Restricciones

- Es responsabilidad del Usuario proporcionar los datos necesarios para la ejecución de las pruebas de certificación.

## 8. Criterios de Aceptación para el Aseguramiento y Control de Calidad del Producto

- Criterios de Aceptación del Aseguramiento de Calidad del Producto
  - Que al final del proyecto no existan hallazgos abiertos.
  - Que todos los requerimientos de negocio definidos en el RET y REF estén especificados de acuerdo al contenido de la Solicitud de Atención.
  - Para la revisión y aceptación de entregables, se considerará que si al menos el 30% de los puntos de verificación contiene incidencias, se devolverá el entregable, y no se considerará como entregado en tiempo y forma, contando como retraso al plan de trabajo.
- Criterios de Aceptación del Control de Calidad del Producto
  - No deben existir defectos abiertos cuya severidad sea Crítica o Alta, o bien que no existan más de 3 defectos de severidad Media.
  - Que se haya generado y aprobado la documentación necesaria de los requerimientos funcionales (Solicitud de

Atención, Especificación Funcional y Especificación Técnica), antes del inicio de las actividades de Control de la Calidad del Producto.

- El alcance del requerimiento está totalmente cubierto por la funcionalidad del producto al concluir las pruebas de Certificación.

- Tablas de Soporte para Parámetros de los Criterios

- Severidad de Hallazgos

La siguiente tabla 5-13 indica una clasificación de severidad de hallazgos en las revisiones de Productos de Trabajo:

Severidad de Hallazgos	Descripción
<b>Alta</b>	<ul style="list-style-type: none"> <li>• Incumplimiento con el objetivo y/o alcance del documento, que cause un impacto negativo en los productos que deriven de él.</li> <li>• No permite que comience, continúe o concluya alguna actividad crítica del proyecto.</li> </ul>
<b>Media</b>	<ul style="list-style-type: none"> <li>• Inconsistencia, errores de redacción que afecten la semántica del documento.</li> <li>• Incumplimiento con el objetivo y/o alcance del documento, que no cause impacto en los productos que deriven de él.</li> <li>• Secciones incompletas u omisiones de contenido sin justificación, que estén incluidas en los rangos severidad Alta o Media.</li> </ul>
<b>Baja</b>	<ul style="list-style-type: none"> <li>• Secciones incompletas u omisiones de contenido sin justificación, que no estén incluidas en los rangos severidad Alta o Media.</li> <li>• Errores de ortografía, gramática o incoherencias de formato; por ejemplo: acentos, puntuación, fuente diferente a la establecida o márgenes.</li> </ul>
<b>Sin Severidad</b>	<ul style="list-style-type: none"> <li>• Cualquier sugerencia de mejora a la documentación.</li> </ul>

Tabla 5-13 Severidad de los Hallazgos (Elaboración Propia)

○ Severidad de los Defectos

A continuación en la tabla 5-14 se muestra una clasificación de severidad de Defectos en las Pruebas de Certificación:

Severidad del Defecto	Descripción
Critica	<ul style="list-style-type: none"> <li>• El sistema no responde y no hay alternativas para ejecutar las pruebas.</li> <li>• El sistema no permite la ejecución del flujo de inicio a fin o bien no hay alternativas para continuar con la ejecución de las pruebas.</li> </ul>
Alta	<ul style="list-style-type: none"> <li>• Inconsistencia con las reglas de negocio y validaciones que definidas para el sistema no permita continuar con el flujo de negocio (cálculos incorrectos, no respeta flujos de procesos).</li> <li>• Incumplimiento con los formatos establecidos para los Layouts que requiera el sistema.</li> <li>• Inconsistencia de datos.</li> <li>• Funcionalidad no implementada, de acuerdo a los requerimientos definidos.</li> </ul>
Media	<ul style="list-style-type: none"> <li>• Defectos en la Interfaz Gráfica.</li> <li>• Errores que permitan continuar con la verificación y/o validación.</li> <li>• Mensajes de error inconsistentes.</li> </ul>
Baja	<ul style="list-style-type: none"> <li>• Errores ortográficos.</li> <li>• Errores de redacción.</li> </ul>
Sin Severidad	<ul style="list-style-type: none"> <li>• Cualquier sugerencia de mejora a la aplicación.</li> </ul>

Tabla 5-14 Severidad de los defectos (Elaboración Propia)

## 9. Documentos para la definición de la Estrategia del Aseguramiento y Control de Calidad del Producto

Los documentos con los que se debe contar previamente, para la

ejecución de las actividades del Aseguramiento y Control de Calidad del Producto se muestran en la tabla 5-15:

Artefacto	Responsable/Organización	Fecha de Entrega	Observaciones
<i>Solicitud Formal de Atención</i>	<i>Empresa Cliente</i>	<i>Sep-1</i>	<i>El documento debe cumplir con el estándar definido la Solicitud de Atención.</i>
<i>Reporte de Especificaciones Funcionales</i>	<i>Empresa Cliente</i>	<i>Oct-1</i>	<i>El documento debe cumplir con el estándar definido en el Reporte de Especificaciones Funcionales.</i>
<i>Reporte de Especificaciones Técnicas</i>	<i>Empresa Cliente</i>	<i>Oct-2</i>	<i>El documento debe cumplir con el estándar definido en el Reporte de Especificaciones Técnicas.</i>

Tabla 5-15 Definición de la Estrategia (Elaboración Propia)

#### 10. Plan de Comunicación

El Plan de Comunicación del presente proyecto establece la línea, por la cual se escalarán todos aquellos riesgos y problemas que impidan la ejecución de las actividades del Aseguramiento y Control de Calidad del Producto, así como los defectos identificados durante la ejecución de las pruebas de certificación.

Siendo 3 los niveles de comunicación establecidos y que comunican a los involucrados en orden ascendente:

- Administrador de Calidad
- Líder de Proyecto
- Jefe del Área de Desarrollo/Jefe de Inteligencia de Negocios
- Gerente de Procesos y Soluciones de Negocio.

## 11. Niveles de Escalamiento

A continuación en la Figura 5-1 se muestran los niveles de escalamiento

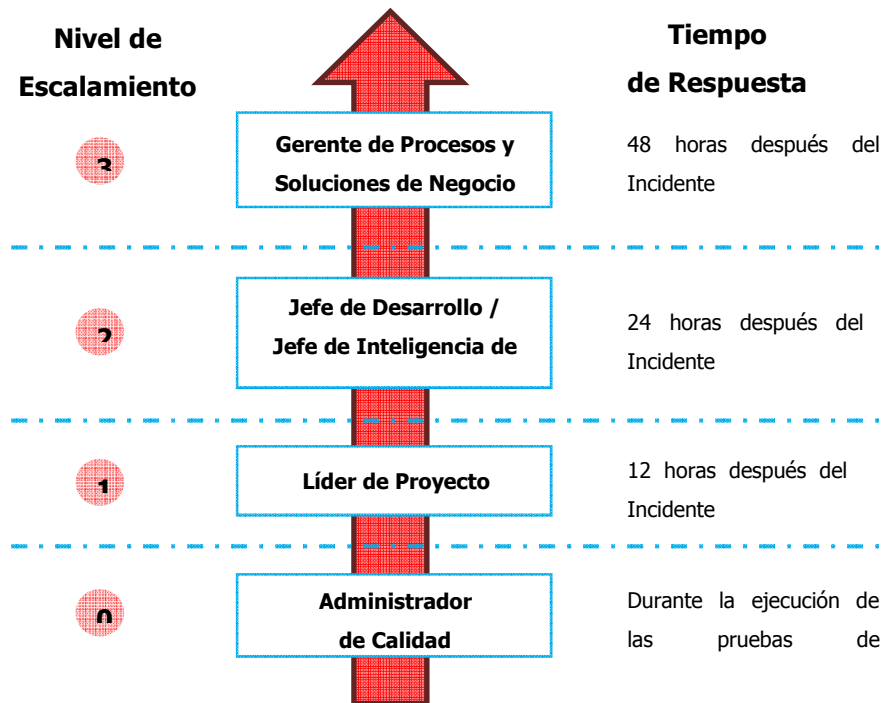


Figura 5-1 Niveles de Escalamiento (Elaboración Propia)

Los entregables como resultado de la ejecución de actividades de Aseguramiento y Control de Calidad del Proyecto son los que se listan en la tabla 5-16:

Artefacto	Nombre del Archivo	Descripción
Listas de verificación de Revisión Funcional Completado.	Check_Revisión_Funcional.xls	Documento que contiene las preguntas a considerar durante la revisión de la especificación funcional.
Listas de verificación de Revisión Técnica Completado.	Check_Revisión_Técnica.xls	Documento que contiene las preguntas a considerar durante la revisión de la especificación funcional.
Plan de Pruebas	Plan_Pruebas.doc	Documento que especifica el alcance y la estrategia de pruebas a realizar.
Casos de Pruebas	MatrizCasosPrueba.xls	Matriz con los casos de Prueba a validar
Evidencia de Pruebas	EvidenciaPrueba.doc	Documento que contiene toda la evidencia de la ejecución de las pruebas de certificación
Reporte de Defectos	ReporteDefectos.xls	Reporte de defectos encontrados durante la ejecución de las pruebas de certificación.

Tabla 5-16 Entregables (Elaboración Propia)

## 12. Riesgos del Proceso de Certificación

A continuación en la tabla 5-17 se muestra los riesgos del proyecto identificados.

Descripción	Impacto	Probabilidad	Acciones	Responsables
No contar con los Casos de Pruebas revisados y aprobados por el Usuario.	Alta	Media	Informar inmediatamente al Líder del Proyecto / Líder Técnico	LL
No contar con una capacitación sobre las funcionalidades impactadas antes del diseño y ejecución de Pruebas.	Media	Media	Informar inmediatamente al Líder del Proyecto / Líder Técnico	CV
No contar con las evidencias de las pruebas realizadas a los productos por el Proveedor de Desarrollo.	Alta	Bajo	Informar inmediatamente al Líder del Proyecto / Líder Técnico	AC
No contar con los ambientes de prueba respectivos.	Alto	Bajo	Informar inmediatamente al Líder del Proyecto / Líder Técnico	DA
No contar con la data de prueba correspondiente.	Alto	Bajo	Informar inmediatamente al Líder del Proyecto / Líder Técnico	AC
No tener accesos a las Bases de Datos de Prueba	Alto	Bajo	Informar inmediatamente al Líder del Proyecto / Líder Técnico	AC

Tabla 5-17 Riesgos del Proceso de Certificación (Elaboración Propia)

## **5.4.2 Análisis**

### **5.4.2.1 Revisión de Requisitos**

A continuación en la tabla 5-18 se muestra los resultados de la revisión de los requisitos del proyecto de Control de Inventarios.



DESCRIPCION	INSTRUCCIONES
ACIERTO	El documento cumple con la verificación
FALLO	El documento no cumple con la verificación
SIN VERIFICAR	No se ha podido verificar si el documento cumple con la verificación
NO APLICA	Para el documento analizado, no aplica la verificación

**PROYECTO:** Control de Inventarios  
**CODIGO PROYECTO:** PRO-0013-0001  
**FASE PROYECTO:** Especificación Funcional  
**DOCUMENTO:** Check\_Requisitos.xls

**FECHA:** ene-02  
**TIPO PROYECTO:** Desarrollo  
**REVISADO POR:** JM  
**VERSION:** 2

*Marcar una X en la casilla que proceda*

Nº	CUESTION	RESULTADO	ID MANTIS	OBSERVACIONES
<b>A</b>	<b>Especificación</b>			
A1	Verificar que se realiza una descripción general de la funcionalidad existente u aspectos generales de negocio sobre la base que se realiza la definición de requisitos.	Acierto		
A2	Verificar que la codificación de la nomenclatura de los requisitos se compone del código del proyecto seguido de un número secuencial, además de una breve título y la descripción que incluya cualquier información necesaria que determine/defina los requisitos y permita su comprensión.	Acierto		
<b>Requisitos Funcionales/Producto</b>				
A3	Verificar que se realiza una lista detallada de los requisitos funcionales o de producto.	Acierto		
A4	Verificar que los requisitos funcionales se definen de forma clara y concisa con objeto de cubrir un solo aspecto.	Acierto		
<b>Requisitos Explotación/Producción</b>				
A5	Verificar que se realiza una descripción de la implementación de los procesos batch conforme a los requisitos de explotación/producción de aplicaciones del cliente.	Acierto		

A6	Verificar que en la descripción de los requisitos de explotación/producción se hace referencia a la nomenclatura del proceso, formato del fichero de log, estructura de directorios, ventana de aplicación, calendario de ejecución, etc.	Acierto		
A7	Verificar que se tiene en cuenta en la descripción del arranque y parada controlada de un nuevo sistema que éstos deben realizarse conforme los requisitos de explotación de aplicaciones del cliente.	Acierto		
A8	Verificar que existe una descripción en las nuevas funcionalidades (requisitos) referente a la comprobación del estado de la aplicación y la validación del sistema a nivel de presencia y actividad, así como componentes y funcionalidad.	Acierto		
A9	Verificar que en algún requisito se indica que, desde el punto de vista de acceso y ejecución de procesos online o batch en el sistema final que soporte la funcionalidad/servicio demandado, el proyecto debe ajustarse a los requisitos de explotación de aplicaciones del cliente	Acierto		
<b>Requisitos Seguridad</b>				
A10	Verificar que el/los requisito/s referente/es a la gestión de usuarios y contraseñas se realizan conforme a los requisitos de seguridad de aplicaciones del cliente.	Acierto		
A11	Verificar que en la descripción de los requisitos sobre la gestión de usuarios y contraseñas se incluyen: - Identificadores de usuario relacionados con la identidad del usuario (nombre y apellidos, por ejemplo). - Longitud de los identificadores de usuario. - No permitir usuarios genéricos. - Limitar el número de intentos. - Longitud y conjunto de caracteres permitidos para las contraseñas. - Periodo de Validez de la contraseña. - No permitir la repetición de contraseñas cuando caducan. - Almacenamiento y transmisión encriptada de contraseñas.	Acierto		
A12	Verificar que se incluye un requisito referente a las copias de seguridad, indicando su periodicidad y recuperación.	Acierto		
A13	Verificar que se incluye un requisito referente al borrado de ficheros temporales, indicando su periodicidad.	Acierto		
A14	Verificar que se incluye un requisito referente a las pruebas con datos reales.	No Aplica		
<b>Requisitos Entregables Software</b>				
A15	Verificar que se incluye un requisito sobre el código fuente del desarrollo, describiendo la entrega del código fuente kernel y no kernel cuando se haya acordado.	Acierto		

A16	Verificar que se realiza un requisito sobre Ejecutables, indicando la entrega de ejecutables, binarios, librerías, etc., tanto kernel como no kernel necesarios para el correcto funcionamiento.	Acierto		
A17	Verificar que se realiza un requisito sobre ficheros auxiliares, indicando la entrega de ficheros auxiliares de configuración, scripts, etc., necesarios para el correcto funcionamiento.	Acierto		
A18	Verificar que se realiza un requisito sobre el entorno servidor, indicando en su descripción de la obligación del correcto funcionamiento de los desarrollos en entorno servidor y con los sistemas operativos del cliente.	Acierto		
A19	Verificar que se realiza un requisito sobre el entorno cliente, indicando en su descripción de la obligación del correcto funcionamiento de los desarrollos sobre las máquinas y con los sistemas operativos del cliente.	Acierto		
<b>Requisitos Entregables Instalación</b>				
A20	Verificar que se realiza un requisito sobre el procedimiento de instalación en los distintos entornos en los que se instale la solución.	No Aplica		
A21	Verificar que se realiza un requisito referente a la responsabilidad del despliegue de la aplicación, tanto a nivel de servidor como de cliente.	No Aplica		
A22	Verificar que se realiza un requisito correspondiente al tiempo requerido de instalación, indicando también todo aquello que implique indisponibilidad del sistema o impacto en otras funcionalidades.	No Aplica		
<b>Requisitos Entregables Documentación</b>				
A23	Verificar que se realiza un requisito acerca de los entregables de documentación mínimos al cliente.	No Aplica		
A24	Verificar que se realiza un requisito acerca de los posibles entregables de documentación adicionales.	No Aplica		
<b>Requisitos Calidad</b>				
A25	Verificar que se realiza un requisito indicando que el análisis y desarrollo del proyecto solicitado debe ser funcionalmente completo, incluyendo las modificaciones necesarias sobre todas las funcionalidades existentes en el sistema o en curso.	No Aplica		
A26	Verificar que se realiza un requisito indicando que todas las cadenas y procesos existentes o en curso, deben seguir funcionando correctamente tras la implantación de la solución propuesta.	No Aplica		
A27	Verificar que se realiza un requisito que indique la mejora de prestaciones del sistema.	No Aplica		
A28	Verificar que se realiza un requisito en el que se contemple la posibilidad de ejecutar pruebas de volumen para la funcionalidad y/o proceso, siempre que sea posible.	No Aplica		
A29	Verificar que se realiza un requisito en el que se plantee una opción con la solución kernel para implementar la funcionalidad solicitada, siempre que sea posible.	No Aplica		

A30	Verificar que se realiza un requisito en el que se indique la intención del proyecto de reducir las necesidades del sistema.	No Aplica		
A31	Verificar que se realiza un requisito en el que se contemple la posibilidad de ejecutar pruebas de volumen para la funcionalidad y/o proceso, siempre que sea posible.	No Aplica		
A32	Verificar que se realiza un requisito en el que se describa la limpieza de objetos obsoletos.	No Aplica		
A33	Verificar que se realiza un requisito en el que se describa la gestión completa de errores, quedando siempre reflejada en el EV la forma de tratar los posibles errores generados.	No Aplica		
A34	Verificar que se realiza un requisito indicando que el proyecto debe funcionar correctamente sobre el entorno de producción, incluyendo la corrección o recuperación ante inconsistencias de datos, configuración del entorno o volumen real de datos	No Aplica		
A35	Verificar que se realiza un requisito en el que se contemple la posibilidad de ejecutar pruebas de volumen para la funcionalidad y/o proceso, siempre que sea posible.	No Aplica		
A36	Verificar que se realiza un requisito que indique la independencia de máquina y alta disponibilidad, con el software de máquina preparado para fail over automático entre máquinas y sin limitaciones funcionales que obliguen a intervenciones manuales.	No Aplica		
A34	Verificar que se realiza un requisito que indique la/as alternativa/as que tenga menor impacto en la disponibilidad del sistema, al hora de la puesta en producción del proyecto.	No Aplica		
A34	Verificar que realiza un requisito que indique la obligatoriedad de no incluir ninguna acción manual de soporte en el proyecto.	No Aplica		
<b>B</b>	<b>Otros</b>			
B1	El documento de Requisitos se ha elaborado en la fase correspondiente del proyecto.	No Aplica		
B2	Los documentos asociados a los requisitos se mantienen actualizados a lo largo del proyecto.	No Aplica		
B3	Se dispone, tras la toma de requisitos, de evidencia de la Revisión de los mismos (RQ actualizado y aprobado en Newark).	No Aplica		
<b>PROPUESTAS MEJORA</b>				
1				
<b>CONCLUSIONES</b>				

Tabla 5-18 Revisión de Requisitos (Elaboración Propia)

### 5.4.2.2 Revisión de Requerimientos Funcionales

A continuación en la tabla 5-19 se muestra el resultado de aplicar el checklist sobre los requerimientos funcionales.

DESCRIPCIÓN	INSTRUCCIONES
ACIERTO	El documento cumple con la verificación
FALLO	El documento no cumple con la verificación
SIN VERIFICAR	No se ha podido verificar si el documento cumple con la verificación
NO APLICA	Para el documento analizado, no aplica la verificación

**PROYECTO:** Control de Inventarios      **FECHA:** dic-01  
**CODIGO**      **TIPO** Desarr  
**PROYECTO:** PRO-0013-0001      **PROYECTO:** ollo  
**FASE**      **REVISADO**  
**PROYECTO:** Especificación Funcional      **POR:** JM  
**DOCUMENTO:** Check\_Revisión\_Funcional.xls      **VERSION:** 2

*Marcar una X en la casilla que proceda*

Nº	CUESTION	RESULTADO	ID MANTI S	OBSERVACIONES
<b>A</b>	<b>Introducción</b>			
A1	Verificar que se incluye un resumen del objetivo de la aplicación.	Acierto		
A2	Verificar que se incluye un resumen de la funcionalidad de la aplicación.	Acierto		
<b>B</b>	<b>Análisis Funcional</b>			
B1	Verificar que se listan las funcionalidades que tendrá que cubrir la aplicación.	Acierto		
B2	Verificar la existencia de la descripción de flujos de negocio.	Acierto		
B3	Verificar la existencia de la descripción de módulos funcionales.	Fallo	56743	Falta describir el módulo de Almacenamiento ABC
B4	Verificar la existencia de la descripción de flujos de pantallas.	Acierto		
<b>C</b>	<b>Configuración</b>			
C	Verificar que se realiza una descripción de parámetros	Acierto		

1	configurables a nivel de negocio.			
C	Verificar que se realiza una descripción de parámetros			
2	configurables a nivel de aplicación.	Acerto		
C	Verificar que se realiza una descripción de perfiles de			
3	usuario.	Acerto		
D	<b>Performance</b>			
D	Verificar que se realiza una definición de los niveles mínimos			
1	de performance de los procesos.	Acerto		
D	Verificar que se incluye una estimación el volumen de			
2	usuarios y datos.	Acerto		
D				
3	Verificar que se incluyen posibles concurrencias.	Acerto		
E	<b>Política de seguridad</b>			
	Verificar que se realiza una definición de niveles mínimos			
E1	respecto a la restricción del acceso a la aplicación.	Acerto		
	Verificar que se realiza una definición de niveles mínimos			
E2	respecto al acceso a los datos.	Acerto		
	Verificar que se realiza una definición de política de backup			
E3	y copias de seguridad.	Acerto		
E4	Verificar que se incluye una política de accesos.	Acerto		
F	<b>Operación y mantenimiento</b>			
	Verificar que se realiza una descripción de la administración			
F1	y configuración de la aplicación.	Acerto		
	Verificar que se realiza una descripción de procesos			
F2	externos y monitorización (descripción, periodicidad...).	Acerto		
	Verificar que se realiza una descripción de la gestión de			
F3	usuarios.	Acerto		
	Verificar que se incluye una descripción de la generación de			
F4	logs y estadísticas.	Acerto		
G	<b>Otros</b>			
E1	El documento de Análisis Funcional se ha elaborado en la			
2	fase correspondiente del proyecto.	Acerto		
E1	Los documentos asociados al diseño se mantienen			
1	actualizados a lo largo del proyecto.	Acerto		
	Se dispone, tras la fase de diseño, de evidencia de la			
E1	Revisión del Diseño (AF/AT actualizados y aprobados en			
2	Newark).	Acerto		
<b>PROPUESTAS MEJORA</b>				
1				
2				
3				
4				

5	
<b>CONCLUSIONES</b>	

Tabla 5-19 Revisión de Requerimientos Funcionales (Elaboración Propia)

#### 5.4.2.3 Revisión de Diseño Técnico

A continuación en la tabla 5-20 se muestra el resultado de aplicar el checklist sobre el documento de especificación técnica.

DESCRIPCION	INSTRUCCIONES
ACIERTO	El documento cumple con la verificación
FALLO	El documento no cumple con la verificación
SIN VERIFICAR	No se ha podido verificar si el documento cumple con la verificación
NO APLICA	Para el documento analizado, no aplica la verificación

**PROYECTO:** Control de Inventarios  
**CODIGO PROYECTO:** PRO-0013-0001  
**FASE PROYECTO:** Especificación Funcional  
**DOCUMENTO:** Check\_Revisión\_Técnica.xls

**FECHA:** ene-02  
**TIPO PROYECTO:** Desarrollo  
**REVISADO POR:** JM  
**VERSION:** 2

Marcar una X en la casilla que proceda

Nº	CUESTION	RESULTADO	ID MANTIS	OBSERVACIONES
<b>A</b>	<b>Introducción</b>			
A1	Verificar que se incluye un resumen del objetivo de la aplicación.	Acierto		
A2	Verificar que se incluye un resumen de la funcionalidad de la aplicación.	No Aplica		
<b>B</b>	<b>Arquitectura</b>			
B1	Verificar que se incluye un diagrama de componentes HW y SW.	Acierto		
B2	Verificar que se realiza una descripción de la arquitectura del sistema.	Acierto		
B3	Verificar que se realiza una descripción de todos los componentes que constituyen la arquitectura.	Acierto		
B4	Verificar que se realiza una descripción de los entornos implicados.	Acierto		
<b>C</b>	<b>Componentes</b>			
C1	Verificar que se realiza una descripción completa de todos los componentes que conforman la aplicación.	Acierto		
<b>D</b>	<b>Procesos</b>			
D1	Verificar que se realiza una descripción completa de los procesos, batch, tiempo real, etc que constituyen el funcionamiento de la aplicación.	Acierto		
<b>E</b>	<b>Integración con otros sistemas</b>			
E1	Verificar que se realiza una descripción de los flujos de entrada y de salida de datos.	Acierto		
E2	Verificar que se realiza una descripción de los formatos de los Inputs y Outputs del sistema provenientes o dirigidos hacia otros sistemas del mapa de sistemas del cliente.	Acierto		
E3	Verificar que se realiza una descripción de procesos vinculados.	Acierto		
<b>F</b>	<b>Otros</b>			
E12	El documento de Análisis Técnico se ha elaborado en la fase correspondiente del proyecto.	Sin Verificar		
E11	Los documentos asociados al diseño se mantienen actualizados a lo largo del proyecto.	Acierto		
E12	Se dispone, tras la fase de diseño, de evidencia de la Revisión del Diseño (AF/AT actualizados y aprobados en Newark).	Acierto		
<b>PROPUESTAS MEJORA</b>				
1				
2				
3				
4				
5				



CONCLUSIONES

Tabla 5-20 Revisión Técnica (Elaboración Propia)

### 5.4.3 Diseño

#### 5.4.3.1 Casos de Pruebas

A continuación en la Tabla 5-21 se muestran los casos de prueba principales utilizados en la ejecución de las pruebas.

ID y Nombre del Requerimiento	09639 – Nuevo Sistema de Control de Inventarios
Líder del Proyecto	Antonio Valcarcel
Administrador de Calidad	Maria Fernanda Valenzuela
Líder Técnico de Calidad	Susana Ancelmo

Versión del REF con la que se genera Diseño	1.0
Versión del RET con la que se genera Diseño	N/A

**Objetivo:**

Verificar que el sistema se comporta tal y como está especificado en la "Especificación Funcional".

ID	Aplicación / Módulo / Componente	Nombre del caso de Prueba (Flujo Básico / Alternativo)	Para Ratificación	Complejidad	Precondiciones	Postcondiciones	Pasos	Puntos de Verificación (PV) / Resultado Esperado	Ciclo 1 de Pruebas de Certificación		
									Periodo de ejecución: Feb		
									Ingeniero de Pruebas: Ana Maria Gonzales		
									Resultado del PV	Resultado del Guión de Prueba	ID Defecto / Comentarios
1	Recepción de Mercaderías	Recepción de Mercaderías tal y como indica la OC.	Sí	Media	Usuario creado con perfil "Operador de Almacén"	Se guarda la información de la recepción.	1. Accede al sistema con el perfil "Operador de Almacén". Login: "usrOperador1" Pass"X328"	1	PASO	PASO	
							2. Elegir la opción "Registrar Recepción de Mercadería"	2. Se muestra interfaz: ALM001.005	PASO		
							3. Registre el número de la	3. El sistema muestra el detalle de la OC definida en	PASO		

							OC y presione el botón "Buscar"	la búsqueda. Verifique que la información cargada es correcta.				
							4. Para cada producto, confirme las cantidad de productos mostrada por defecto por el sistema (coincide con la OC original).	4.	PASO			
							5. Presione el botón "Registrar Recepción".	5. El sistema muestra el mensaje "Recepción de OC [999] se guardó exitosamente"	PASO			
							6. Utilice el CP "Verificar Stock"	6. Verifique el aumento del stock de las mercaderías.	PASO			
2	Sistema de Control de Inventarios - Acceso	Login Satisfactorio	si	baja	Usuario Creado	1.Ingresar a la url: CtrlInv.aspx	1Verificar que levante el sistema	PASO	PASO			
						2.Ingresar usuario Valido	2 Verificar que se muestre un check verde al lado del txtbox	PASO				
						3.Ingresar PAssword Valido	3 Verificar que se muestre un check verde al lado del txtbox	PASO				
						4.Click en Ingresar	4 Verificar el acceso satisfactorio al sistema	PASO				
3	Sistema de Control de Inventarios - Acceso	Login – usuario erroneo	si	baja	Usuario Creado	1.Ingresar a la url: CtrlInv.aspx	1Verificar que levante el sistema	PASO	PASO			
						2.Ingresar usuario	2 Verificar que se muestre una	PASO				

						invalido	"X" roja al lado del txtbox				
						3.Ingresar Password Valido	3 Verificar que no permita ingresar Password	PASO			
						4.Click en Ingresar	4 Verificar que el botón se encuentre deshabilitado	PASO			
4	Módulo de Planeamiento de Requerimiento de Materiales	Pedidos Automáticos / Configuración	si	media	Acceso al modulo	1.Registrar información sobre: -Producto -Inventario inicial -Inventario Mínimo -Inventario Máximo -Lead Time -Proveedores	1. Verificar que el sistema acepte la información por cada uno de los parámetros solicitados.	PASO	PASO		
						2.Click en guardar configuración	2. Verificar que la configuración haya sido guardada a nivel de base de datos.	PASO			
						3.Click en Habilitar la configuración	3. Verificar que se haya habilitado la configuración sin problemas	PASO			
5	Módulo de Planeamiento de Requerimiento de Materiales	Pedidos Automáticos / Pedido realizado correctamente	si	media	Habilitación de la Configuración de pedido	1. Modificar por base de datos los parámetros requeridos para el pedido	1. Verificar que se muestre un mensaje en pantalla indicando la realización del	PASO	FALLO		

						automático	pedido				
							2.Verificar en el log de Pedidos que se hayan registrado las cantidades correspondientes	FALLO		001	
							3.Verificar que se hayan enviado los mails respectivos a los proveedores	FALLO		002	
						2.Aceptar el pedido realizado	2. Verificar que el foco vuelva al menú principal.	PASO			
6	Módulo de Planeamiento de Requerimiento de Materiales	Pedidos Automáticos / Pedido no realizado	si	media	Habilitación de la Configuración de pedido	1. Modificar por base de datos los parámetros requeridos para el pedido automático, excepto uno.	1. Verificar que no se muestre ningún mensaje	PASO	PASO		
							2.Verificar que el log de Pedidos no haya sufrido cambios	PASO			
							3.Verificar que no se hayan enviado los mails respectivos a los proveedores	PASO			
7	Módulo de Planeamiento de Requerimiento de Materiales	Control de Perecibles / Reposición	si	media	Productos registrados con el detalle requerido	1. Mover la fecha del sistema a la fecha perecible	1 Verificar mensaje por pantalla informando que	PASO	FALLO		

						de algún producto.	el producto está próximo a perecer.				
							2.Verificar que se solicite la reposición o retiro del producto	PASO			
						2.Aceptar la reposición	1. Verificar que los correos a los proveedores hayan sido enviados	FALLO		003	
							2.Verificar el registros de los pedidos en base de datos	FALLO		004	
							3. Verificar que se hayan registrado las entradas en el log de pedidos.	FALLO		005	
8	Módulo de Planeamiento de Requerimiento de Materiales	Control de Perecibles / retiro de Productos	si	media	Productos registrados con el detalle requerido	1. Mover la fecha del sistema a la fecha perecible de algún producto.	1 Verificar mensaje por pantalla informando que el producto está próximo a perecer.	PASO			
							2.Verificar que se solicite la reposición o retiro del producto	PASO	PASO		
						2.Aceptar el retiro del producto	1. Verificar que el producto haya sido marcado para retiro	PASO			
9	Almacenaje	Módulo de	si	compleja	Mapa	1.Ingresar un		PASO			

		Organización de Anaqueles – Posición Actual			cargado	nuevo lote al inventario					
						2.Seleccionar Posición en anaquel	1. Verificar que el sistema muestre la posición actual.	PASO			
							2.Verificar que se muestre el botón "Aceptar Posición Actual" y "Modificar Posición"	PASO			
						3.Seleccionar posición actual	1. Verificar que el sistema guarde la posición con el nuevo lote ingresado.	PASO			
10	Almacenaje	Módulo de Organización de Anaqueles – Modificar Posición	si	compleja	Mapa cargado	1.Ingresar un nuevo lote al inventario		PASO	PASO		
						2.Seleccionar Posición en anaquel	1. Verificar que el sistema muestre la posición actual.	PASO			
							2.Verificar que se muestre el botón "Aceptar Posición Actual" y "Modificar Posición"	PASO			
						3.Seleccionar Modificar Posición	1. Verificar que el sistema guarde la posición con el nuevo lote ingresado.	PASO			

11	Almacenaje	Módulo de Control RFID – Carga por lotes	si	compleja	RFID Configurado	1.Cargar inventario por lotes al almacén por medio de la lectora RFID	1.Verificar que la cantidad de inventario se cargado se muestre correctamente	PASO	PASO		
							2.Verificar que los registros cargados estén completos en base de datos	PASO			
12	Almacenaje	Módulo de Control RFID – Carga por unidades	si	compleja	RFID Configurado	1.Cargar inventario por unidades al almacén por medio de la lectora RFID	1.Verificar que la cantidad de inventario se cargado se muestre correctamente	PASO	PASO		
							2.Verificar que los registros cargados estén completos en base de datos	PASO			
13	Generación de Reportes de Control	Verificar la cantidad de mermas x mes	si	Media	Acceso al módulo de reportes	1.Ingresar al módulo de reportes		PASO	FALLO		
						2.Emitir el reportes de mermas para el mes anterior	1. Verificar que se haya emitido correctamente, contrastando las cantidades a nivel de base de datos.	FALLO		006	
14	Generación de Reportes de Control	Productos más vendidos	si	Media	Acceso al módulo de reportes	1.Ingresar al módulo de reportes		PASO			
						2.Emitir el reportes de Productos más	1. Verificar que se haya emitido correctamente,	FALLO		007	



						vendidos	contrastando las cantidades a nivel de base de datos.			
15	Generación de Reportes de Control	Solicitudes Cumplidas	no	si	Media	Acceso al módulo de reportes	1.Ingresar al módulo de reportes	PASO		
						2.Emitir el reportes de Productos más vendidos	1. Verificar que se haya emitido correctamente, contrastando las cantidades a nivel de base de datos.	FALLO		008

Tabla 5-21 Registro de Casos de Prueba (Elaboración Propia)

#### **5.4.4 Pruebas y Ratificación**

##### **5.4.4.1 Ejecución de Pruebas**

Después de que el desarrollo del sistema de control de inventarios se concluyó, se inició la ejecución de las pruebas. Este punto se realizó después de haber revisado el correcto funcionamiento del ambiente de pruebas y haber obtenido la data con la que se realizaron las pruebas. La ejecución de los casos de pruebas se realizó 3 veces hasta que se lograron levantar todas las incidencias.

La Figura 5-22 muestra los defectos encontrados durante las pruebas.

ID y Nombre del Requerimiento	09639 – Nuevo Sistema de Control de Inventarios
Líder del Proyecto	Antonio Valcarcel
Administrador de Calidad	Maria Fernanda Valenzuela
Líder Técnico de Calidad	Susana Ancelmo

Registro de Defecto											Acciones a Realizar			Seguimiento		
ID	Ciclo de prueba	Aplicación	Módulo	caso de Prueba	Descripción del Defecto	Fecha de Registro	Severidad	Prioridad	Estado	Tipo	Acción Correctiva	Responsable	Fecha Compromiso	Fecha de Solución	Fecha de Seguimiento	Observaciones
001	1	Control de Inventarios	Pedidos Automatizados	4	El log de pedidos no muestra información completa	02/02/2015	media	media	Cerrado	Código	Modificar el procedures de registro de log	Lexter Sanchez	04/02/2015	03/02/2015	04/02/2015	
002	1	Control de Inventarios	Pedidos Automatizados	4	Los mails no fueron enviados a los proveedores	02/02/2015	media	media	Cerrado	Configuración	Revisar la configuración del servidor de correos	Gary Caceres	04/02/2015	03/02/2015	04/02/2015	
003	1	Control de Inventarios	Control de Percibiles	6	Los mails no fueron enviados a los proveedores	02/02/2015	media	media	Cerrado	Configuración	Revisar la configuración del servidor de correos	Gary Caceres	04/02/2015	03/02/2015	04/02/2015	
004	1	Control de Inventarios	Control de Percibiles	6	No se han llenado todos los campos necesarios en base de datos	03/02/2015	Media	media	cerrado	Código	Modificar el procedures de registro de Base de datos	Lexter Sanchez	05/02/2015	04/02/2015	05/02/2015	
005	1	Control de Inventarios	Control de Percibiles	6	No se han registrado todas las entradas al log	03/02/2015	Media	media	cerrado	Código	Modificar el procedures de registro de log	Lexter Sanchez	05/02/2015	04/02/2015	05/02/2015	
006	1	Control de Inventarios	Reportes	12	Las cantidades	04/02/2015	media	media	cerrado	Código	Modificar la fuente de	Susana Rebaza	05/02/2015	05/02/2015	05/02/2015	

					mostradas en el reporte no coinciden con base de datos						obtención de los Reports					
007	1	Control de Inventarios	Reportes	13	Las cantidades mostradas en el reporte no coinciden con base de datos	04/02/2015	media	media	cerrado	Código	Modificar la fuente de obtención de los Reports	Susana Rebaza	05/02/2015	05/02/2015	05/02/2015	
008	1	Control de Inventarios	Reportes	14	Las cantidades mostradas en el reporte no coinciden con base de datos	04/02/2015	media	media	cerrado	Código	Modificar la fuente de obtención de los Reports	Susana Rebaza	05/02/2015	05/02/2015	05/02/2015	

Tabla 5-22 Registro de Defectos (Elaboración Propia)

## 5.5 Estadísticas y Métricas

### 5.5.1 Defectos por Módulo.

Esta medición identifica los módulos de la aplicación con mayor cantidad de defectos.

El análisis de este grafico permite que el líder de certificación cambie el esfuerzo dedicado a los diferentes módulos de la aplicación, tomando como criterio los módulos que presentan la mayor cantidad de defectos. Esto permite que el equipo de certificación halle la mayor cantidad de defectos tempranamente y centre sus esfuerzos en los módulos más problemáticos. Las mediciones son mostradas en un diagrama de Pareto (Ver Figura 5-2) para mejorar la identificación de los módulos con mayor densidad de defectos. El grafico muestra el resultado final luego de aplicar el proceso propuesto en el presente trabajo.

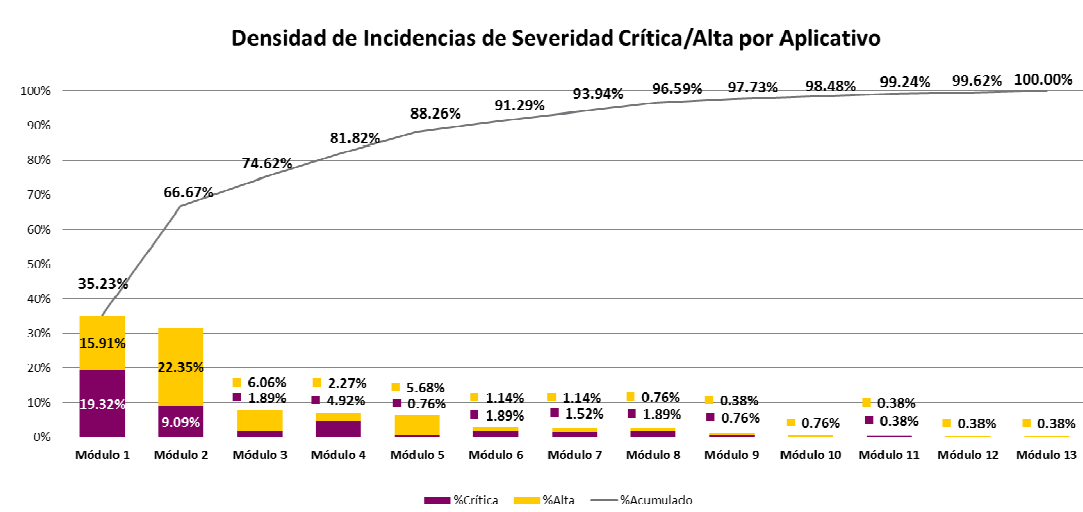


Figura 5-2 Densidad de Incidencias por Aplicativo (Elaboración Propia)

### 5.5.2 Numero de caminos probados.

Luego de aplicar la fase de pruebas se ha calculado que se ha probado un 82% del total de caminos existentes en la aplicación. Los datos de la industria muestran que las pruebas ad hoc logran entre el 60% y el 75% de la cobertura de caminos probados. Las revisiones ayudaron a mejorar los entregables de desarrollo y por lo tanto indirectamente mejorar los casos de pruebas y la cobertura de los caminos.

$$\text{Número de caminos probados} = \frac{\text{cantidad de caminos probados}}{\text{cantidad total de caminos}} = 82\%$$

#### 5.5.3 Efectividad de la detección de defectos.

A la finalización del presente trabajo no existen incidencias registradas durante el uso del software en el ambiente de producción. Lo anterior se debe al reciente despliegue de la aplicación en producción.

$$= \frac{\text{defectos hallados en las pruebas}}{\text{defectos hallados en las pruebas} + \text{defectos hallados en producción}}$$

#### 5.5.4 Presupuesto utilizado en revisiones.

El presupuesto invertido en las actividades de revisión alcanzó un 12% del presupuesto total asignado al proyecto. Las estadísticas de la industria definen que las revisiones consumen entre el 5% y el 15% del esfuerzo de desarrollo.

$$\text{Presupuesto Utilizado en Revisiones} = \frac{\text{presupuesto revisiones}}{\text{presupuesto total del proyecto}} = 12\%$$

### 5.5.5 Incidencias Por Tipo de Revisiones y Pruebas.

La figura 5-3 muestra la cantidad de incidencias identificadas para cada uno de los tipos de pruebas y revisiones hallados al proyecto de Control de Inventarios.

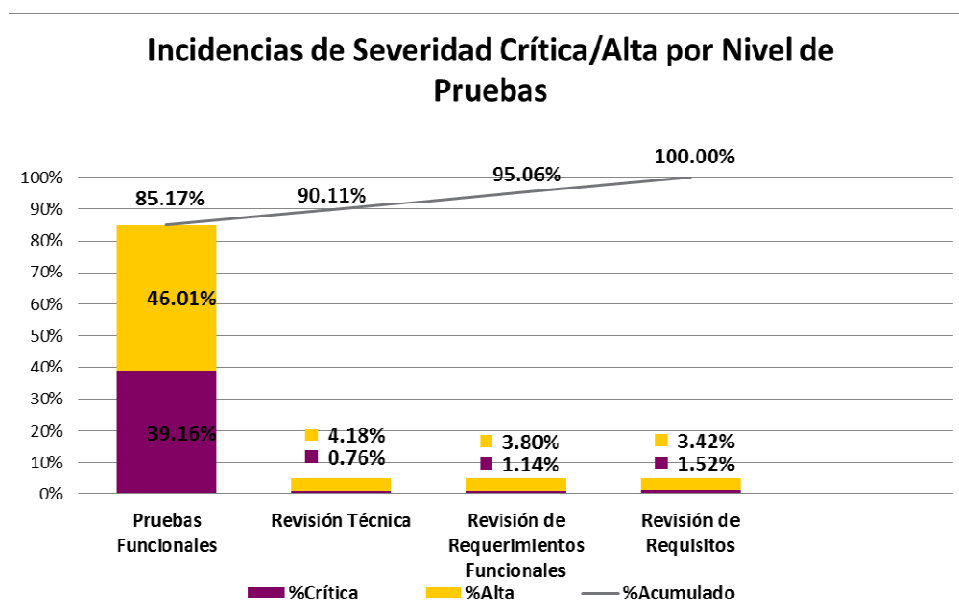


Figura 5-3 Incidencias Por Tipo de Revisiones y Pruebas (Elaboración Propia)

## 5.6 Costos y tiempo.

Debido a que fue el primer caso de estudio para ver la aplicación del Marco Metodológico propuesto para las actividades del Proceso de Verificación y Validación, el costo incurrido por la Pyme en estudio fue cero, sin embargo, si más adelante otra Pyme decide mejorar sus procesos de calidad y adoptar este Marco Metodológico propuesto se sugiere un costo de S/.3.500.

A continuación en la siguiente Tabla 5-23 se muestra una propuesta de costos y la duración de la capacitación y asesorías en los que se podría incurrir.

Costos	Duración de Capacitación y Consultoría
S/. 3,500.000	5 días

Tabla 5-23 Propuesta de Costos y Duración de Capacitación y Asesorías (Elaboración Propia)

## 5.7 Resultados obtenidos.

A diferencia de otros proyectos ejecutados por la empresa que tenían la misma envergadura y que eran muy similares al caso de estudio, el **Proyecto de Sistema de Control de Inventarios** se concluyó dentro del tiempo y costos estimados inicialmente. Después de realizar el pase a Producción, se llegaron a detectar incidentes pero no fueron críticos, esto ocurrió porque el ambiente de pruebas en el cual se realizó la certificación del proyecto no tenía las mismas características que el ambiente de Producción. Lo más importante es que el cliente de la empresa se mostró satisfecho con los resultados obtenidos.

Por otro lado, los empleados que participaron en el proyecto se mostraron más comprometidos con la implementación de este proyecto, principalmente porque sabían qué procesos ejecutar, el momento en el que deberían hacerlo y las métricas que deberían utilizar.

Finalmente debido a la mejora obtenida, la empresa decidió adoptar este Marco Metodológico relacionado a las actividades de Verificación y Validación en la implementación de futuros proyectos de desarrollo.

En el presente Capítulo V se realizó una descripción Marco Metodológico propuesto del proceso de Verificación y Validación de Software, en el siguiente Capítulo VI se mostrará las conclusiones y trabajos futuros.



## **Capítulo 6: Conclusiones y Trabajos Futuros**

En el presente capítulo se describen las conclusiones y los trabajos futuros que pueden ser llevados a cabo por otra investigación y que pueden tomar como referencia el presente trabajo de investigación.

### **6.1 Conclusiones**

1. Se desarrolló el Marco Metodológico para el proceso de Verificación y Validación de software ajustado a la realidad socio económico de una Pyme basado en los modelos y estándares de calidad más representativos del mercado. Asimismo, se incluyó una guía metodológica detallada, los roles y competencias del equipo de trabajo, las principales métricas y las plantillas/listas de verificación base que deben ser considerados durante la implementación de un Software de Calidad.
2. Se realizó un estudio de los principales modelos, estándares, normas y estudios relacionados al Proceso de Verificación y Validación.
3. Se realizó comparaciones entre de los Modelos de Calidad y se determinaron qué ventajas, deficiencias presentan para ser aplicados en una Pyme.
4. Se definió un conjunto de roles y competencias que debería tener el equipo de trabajo para asegurar la correcta ejecución de los procesos de Verificación y Validación.
5. Se definió las principales métricas y las plantillas/listas de verificación que sirvieron como base para la ejecución de los procesos de Verificación y Validación de Software.

6. Se aplicó el Marco Metodológico propuesto durante la implementación del Sistema de Control de Inventarios de la Pyme QBOSS SAC.
7. Después de aplicar el Marco Metodológico la empresa QBOSS obtuvo una mejora importante en sus procesos de implementación de Software.
8. Involucrar al personal y obtener buenos resultados en un periodo de tiempo reducido, fue un aspecto muy importante para aumentar la motivación del personal.
9. La comunicación constante y la participación activa de todos los empleados implicados en la implantación del Marco Metodológico fue también un aspecto clave para el éxito del caso de estudio.

## **6.2 Trabajos Futuros**

1. En el presente trabajo sólo contempla las plantillas para la Verificación y Validación estática de los tres documentos más utilizados en la Pymes: Especificación de Requisitos, Especificación Funcional y Técnica, otros trabajos de Investigación podrían incluir plantillas para diagramas de arquitectura u otros documentos de interés.
2. La presente Tesis incluye la Verificación y Validación dinámica mínima que debe realizar un Pyme, los trabajos de investigación futuros podrían incluir una guía para la automatización o pruebas de Performance, Seguridad o Usabilidad.
3. El Marco Metodológico propuesto sólo incluye una guía para la implementación de Software convencional, otras investigaciones podrían

incluir una guía completa enfocadas en metodologías ágiles como XP (Extreme Programming), FDD (Feature-driven development) y Scrum.

## Referencias Bibliográficas

- (ABRAN, A., MOORE, J.W., BOURQUE, P., DUPUIS, R., and TRIPP, L.L. 2004). SWEBOOK: Guide to the Software Engineering Body of Knowledge: 2004 version. IEEE Computer Society, Los Alamitos, CA, Tokyo.
- (ALLUÉ, ALBERTO; DOMÍNGUEZ, ELADIO; LÓPEZ, ANTONIO; ZAPATA, MARÍA A., 2013). QRP: a CMMI Appraisal Tool for Project Quality Management. CENTERIS 2013 - Conference on ENTERprise Information Systems / PROjMAN 2013 - International Conference on Project MANagement / HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies.
- (ARBOLEDAA, HUGO; PAZBY RUBBY CASALLAS, ANDRÉS, 2013). Metodología para implantar el Modelo Integrado de Capacidad de Madurez en grupos pequeños y emergentes. Universidad ICESI.
- (BACH, 1999) Heuristic risk-based testing, Software Testing and Quality Engineering Magazine 11.
- (BAKER. C., 1957). Review of D.D. McCracken's "Digital Computer Programming". Mathematical Tables and Other Aids to Computation.
- (BATISTA, J Y A. FIGUEREIDO, 2000). SPI in a very small team: a case with CMM. Software process: Improvement and practice Vol5.
- (BEIZER. B., 1984). Software Testing and Quality Assurance. Van Nostrand Reinhold Co., New York, NY, USA.
- (BEIZER. B., 1990). Software System Testing Techniques Second Edition. Van Nostrand Reinhold Co., New York, NY, USA.
- (BELT, P., HARKONEN, J., MOTTONEN, M., KESS, P., and HAAPASALO, H., 2008). Improving the efficiency of verification and validation. International Journal of Services and Standards.
- (BIRD. D and Muñoz. D, 1982) "Automatic generation of random selfchecking test cases" IBM.
- (BLACK, REX, 2017) Pragmatic Software Testing: Becoming an effective and efficient test professional.
- (BOEHM B., 1979) W. Guidelines for Verifying and Validating software requirements and design specifications Technical report, CA, USA.
- (BOYER. R, ELPAS. B, LEVITT. K, 1975). "A formal system for testing and debugging programs by symbolic execution".

- (BURNSTEIN I., 2003). Practical Software Testing. Springer-Verlag: Chicago, USA.
- (BURNSTEIN, I., SUWANNASART, T., CARLSON C.R., 1996a). Developing a Testing Maturity Model: Part I. CrossTalk, STSC, Hill Air Force Base.
- (BURNSTEIN, I., SUWANNASART, T., CARLSON C.R., 1996b) Developing a Testing Maturity Model: Part II. CrossTalk, STSC, Hill Air Force Base.
- (CALVO MANZANO, 1999) Metodos de mejora del proceso de desarrollo de sistemas de información en la pequeña y mediana empresa.
- (CLARKE, 1976), "A system to generate test data and symbolically execute programs". IEEE.
- (CMMI-DEV, 2010), CMMI for Development version 1.3. Software Engineering Institute (SEI), Pittsburg, USA.
- (DEMING, W. EDWARD, 1986); "Out of the Crisis"; Cambridge, MA: MIT. Center for Advanced Engineering. (ISBN 0- 262-54115-7)
- (DEUTSCH, M.,1979)"Verification and Validation", Software Engineering, R.Jensen y C. Tonies(eds.) Prentice Hall
- (DRIEL, 2010) Ewout van Driel, "Software Control y Testing TMap Next® Pruebas impulsadas por los objetivos de negocio", SOGETI España.
- (FRANK ELBERZHAGER JÜRGEN MÜNCH; VI TRAN NGOC NHA, 2012). A systematic mapping study on the combination of static and dynamic quality assurance techniques.
- (GOSLIN, A. 2009). TMMi Assessment Method Application Requirements (TAMAR). TMMi Foundation: Ireland.
- (HARETON, 2001) A process framework for small projects. Software Process: Improvement and practice Vol.6.
- (HETZEL, B.,1991) "Software Testing: Some Troubling Issues and Opportunities", BCS Special Interest Group in Software Testing, Dec 6, 1991.
- (HOWDEN W.E., 1976) "Reliability of the Path Analysis Testing Strategy". IEEE Transactions on Software Engineering.
- (HOWDEN W.E., 1978) Theoretical and Empirical Studies of Program Testing. IEEE Transactions Software Engineering.
- (ANSI/IEEE 829,1983) Standard for Software Verification and Validation. IEEE Computer Society, Software Engineering Standard Committee.
- (ANSI/IEEE 1008-1987) Estándar basado en las pruebas unitarias. IEEE Computer

Society, Software Engineering Standard Committee.

(IEEE 1012, 2004) Standard for Software Verification and Validation. IEEE Computer Society, Software Engineering Standard Committee.

(IEEE 1028, 2008), Standard for Software Reviews and Audits. IEEE Computer Society, Software Engineering Standard Committee.

(IEEE 730, 2002), Standard for Software Quality Assurance Plans. IEEE Computer Society, Software Engineering Standard Committee.

(IEEE 829, 1998) Software test documentation. IEEE Computer Society, Software Engineering Technical Committee.

(IEEE/ANSI 610.12, 1990), IEEE Standard Glossary of Software Engineering Terminology. IEEE Computer Society, Software Engineering Standard Committee.

(IEEE1012, 2004) Standard for Software Verification and Validation, IEEE 1012, 2004.

(INCE.D, 1987) "The automatic generation of Testing Data" Comput J, 30.

(ISO/IEC 12207:1995) Software Testing Standard

(ISO/IEC 29119, 2009) Software Testing Standard:  
<http://www.softwaretestingstandard.org/>

(ISO/IEC 29110) Software Testing Standard

(KIT, E., 1995). Software Testing in the Real World: Addison-Wesley: Gran Bretaña.

(KOOMEN, T., VAN DER AALST, L., BROEKMAN, B., VROON, M., 2006). TMap Next for result-driven testing. UTN Publishers: Netherlands.

(KOREL B., 1990). Automated test data generation IEEE Transactions on Software Engineering.

(LEE, C. 2009). Adapting and adjusting test process reflecting characteristics of embedded software and industrial properties based on referential models. In the Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, (pp. 1372-1377), Seoul, Republic of Korea. ACM.

(LUNA, SERGIO; LOPES, AMIT; YAN SEE TAO, HOONG; ZAPATA, FRANCISCO; PINEDA, RICARDO, 2013). Integration, Verification, Validation, Test, and Evaluation (IVVT&E) Framework for System of Systems (SoS). Conference Organized by Missouri University of Science and Technology 2013- Baltimore, MD.

(LUNA, SERGIO; LOPES, AMIT; YAN SEE TAO, HOONG; ZAPATA, FRANCISCO; PINEDA, RICARDO, 2013). Developing an analytical model for planning systems verification, validation and testing processes. Conference Organized by Missouri

University of Science and Technology 2013- Baltimore, MD.

(MARTIN, 2002) "Software testing : a guide to the TMap approach", Editor Addison-Wesley, ISBN 9780201745719.

(MILLER. E. F., 1978). Program Testing – An overview for managers. In the Proceedings of the IEEE Computer Society's Second International Computer Software and Applications Conference, Chicago, USA. IEEE.

(MILLER. E. HOWDEN.W.E.,1981). Software Testing and Validation Techniques. IEEE Computer Society Press: New York, EEUU.

(MISHRA, D., MISHRA, A. 2007). Efficient software review process for small and medium enterprises. IET Software.

(MYERSG F., 1979). Art of Software Testing. John Wiley & Sons, Inc.: New York, EEUU.

(OKTABA, H.; PIATTINI, M.; PINO, F.; OROZCO, M.J; ALQUICIRA, 2008), E. COMPETISOFT. "Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos". Madrid España, RA-MA Editorial.

(PFLEEGER S.,2001) "Software Engineering, 2nd Edition", ISBN: 0130290491,Prentice Hill.

(PRESSMAN, ROGER S., 2001), Software Engineering. A practitioner's Approach, Quinta Edición.

(RAMAMOORTHY,C., Ho, S., AND CHEN, W., 1976), "On the automated generation of program test data". IEEE Trans. Softw. Eng.

(RON BURBACK, 1998) Software Engineering Methodology.

(SAEDIAN, 1979) Characterizing a software process maturity model for small organizations. ACM SIGICE bulletin Vol.23

(SANCHEZ, F., SANCHO, M.R., BOTELLA, P., GARCIA, J., ALUJA, T., NAVARRO, J., and BALCAZAR, J.L., 2008). In the Proceedings of the Jornadas de Enseñanza Universitaria de la Informatica (JENUI 2008), (pp. 123-130), Granada, Spain.

(SEI, 2010). CMMI® for Development (CMMI-DEV) version 1.3. Software Engineering Institute, Pittsburg, USA.

(SHEWHART W., 1931). Economic Control of Quality of Manufactured Product. American Society for Quality: Milwaukee, USA.

(SOGETI, 2008a). TPI home pages. Accesed 6th May 2008 at <http://www.sogeti.nl/Home/Expertise/Testen/TPI.jsp>.

(SOGETI, 2008b). Structured Test Process. Accesed 14th May 2008 at

[http://eng.tmap.net/Home/TMap/The\\_4\\_essentials/Structured\\_Test\\_Process.jsp](http://eng.tmap.net/Home/TMap/The_4_essentials/Structured_Test_Process.jsp).  
(SOGETI, 2008c). Complete Tool Box. Accessed 14th May 2008 at [http://eng.tmap.net/Home/TMap/The\\_4\\_essentials/Complete\\_tool\\_box.jsp](http://eng.tmap.net/Home/TMap/The_4_essentials/Complete_tool_box.jsp).

(SOGETI, 2008d). Adaptive. Accessed 20th May 2008 at [http://eng.tmap.net/Home/TMap/The\\_4\\_essentials/Adaptive.jsp](http://eng.tmap.net/Home/TMap/The_4_essentials/Adaptive.jsp).

(SOMMERVILLE, IAN, 2005), Software Engineering 07 Edition.

(STEINER, M., BLASCHKE, M., PHILIPP, M., and SCHWEIGERT, T. , 2010) Make test process assessment similar to software process assessment—the Test SPICE approach. Journal of Software Maintenance and Evolution: Research And Practice, early view. DOI: <http://dx.doi.org/10.1002/smr.507>.

(STEINER, M., BLASCHKE, M., PHILIPP, M., SCHWEIGERT, T. 2010). Make test process assessment similar to software process assessment—the Test SPICE approach. Journal of Software Maintenance and Evolution: Research And Practice, early view, DOI: <http://dx.doi.org/10.1002/smr.507>.

(SWEBOK, 2004) Software Engineering Body of Knowledge

(TASSEY, G. 2002). The economic impacts of inadequate infrastructure for software testing. Technical Report RTI 7007-011.

(TIAN, T, 2005). Software Quality Engineering: Testing. John Wiley & Sons: New York.

(TMAP, 2010) “Essentials of TMap Next”, [http://eng.tmap.net/Home/TMap/The\\_4\\_essentials/index.jsp](http://eng.tmap.net/Home/TMap/The_4_essentials/index.jsp)

(TMAP, 2014) TMAP - BASED PROJECT AND PROGRAM TESTING <http://www.sogeti.com/solutions/testing/tmap/> (Consultado 16-5-2015).

(TOKAR R., MANKEFORTS S., 2003). A survey on testing and reuse. In the Proceedings of the IEEE International Conference on Software – Science, Technology and Engineering (SwSTE'03), Herzelia, Israel. IEEE Computer Society.

(TURING. A., 1950) Computing Machinery and Intelligence. Mind a quarterly review of psychology and philosophy.

(VATESH PASRIJA, SANJAY KUMAR, PRAVEEN RANJAN SRIVASTAVA, 2012). Assessment of Software Quality: Choquet Integral Approach. Computer Science & Information System Department.

(VEENENDAAL, E. 2010). Test Maturity Model integration (TMMi). TMMi Foundation, Ireland.

(VEENENDAAL, E., 2010) Test Maturity Model integration (TMMi). TMMi Foundation, Ireland.



(VEENENDAAL, E., BATH, G., EVANS. I. 2010). Expert Level Syllabus Improving the Testing Process. Accessed 7th September at [http://istqb.org/download/attachments/2326555/Expert-Level-Syllabus-Improving-TheTestProcess-Release-V1\\_0\\_2.pdf](http://istqb.org/download/attachments/2326555/Expert-Level-Syllabus-Improving-TheTestProcess-Release-V1_0_2.pdf).

(VINICIUS HUMBERTO SERAPILHA DURELLIA; RODRIGO FRAXINO ARAUJOA; MARCO AURELIO GRACIOTTO SILVA; RAFAEL ALVES PAES DE OLIVEIRAA; JOSE CARLOS MALDONADOA; MARCIO EDUARDO DELAMAROA, 2012). A scoping study on the 25 years of research into software testing in Brazil and an outlook on the future of the area.

(ZAHARAN, 1998) Software Process Improvement. Practical guidelines for business success. Addison – Wesley.